The Role of Teaching Assistants in Introductory Programming Courses

Ayshwaya Saktheeswaran

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Applications

Steve R. Harrison, Chair
Stephen H. Edwards
Deborah G. Tatar

June 30th, 2016
Blacksburg, Virginia

Keywords: computer science education, Teaching Assistants, introductory programming courses

Copyright 2016, Ayshwarya Saktheeswaran
The Role of Teaching Assistants in Introductory Programming Courses

Ayshwarya Saktheeswaran

(ABSTRACT)

The department of computer science, across many schools in the US, have been seeing a constant increase in enrollments over the last decade. This particularly impacts class sizes of introductory courses, as they are usually listed as required prerequisites courses. The students in these courses typically have very little or zero prior experience with programming. In such course settings, it is not an exaggeration to state that most students in these courses spend more one-on-one time with the course teaching assistants than they do with their course instructors. This implies that the kind of individual attention provided by the TAs of such courses to their students has a very high impact on the students’ learning and the quality of the TAs would greatly impact the quality of the course and directly or indirectly also impact the student retention rate and their interest in computer science for their academic/industry careers.

We wanted to take a closer look at what it is that these TAs do, and how they do it. We observed TAs from two introductory courses for almost about two semesters, and conducted a focus group meeting each with TAs and students enrolled in these course. We found that the TAs felt responsible for instilling an interest in computer science in the students, apart from helping them to learn by themselves. We also found that the students see teaching assistants as a very valuable resource, when it comes to actually applying the concepts that they learn in lecture. Our findings tells us that there is a gap between what the TAs think they need to give as help to the students and what the students tend to expect from their TAs. We also discuss the implications of our findings and possible future work.
Acknowledgments

I would like to thank my advisors, Steve R. Harrison, Stephen H. Edwards and Deborah Tatar for providing me with excellent suggestions, improvements, and guidance throughout the process. I really appreciate your prompt response to my numerous emails and your willingness to meet with me whenever I needed help.

I would also like to thank my family for their encouragement, and support throughout my time in grad school. A shout out to Sharon kinder-potter, who has been a friendly face since day 1 of grad school and without whom I would have drowned in enormous amount of paperwork.

Finally, I thank all my friends and lab members who were always there for me when I needed to ramble about my research. A special thanks to Jo, who made sure I was well fed the entire time I spent writing this document.
Contents

1 Introduction .............................. 2
   1.1 Motivation ................................................. 3
   1.2 Research goal ............................................ 4
   1.3 Background ............................................. 5
   1.4 Approach ................................................. 5
   1.5 Organization of this document ......................... 6

2 Literature Review ........................ 7
   2.1 TAs for effective course management .................... 7
   2.2 Training Teaching Assistants .............................. 9
   2.3 Teaching methods ........................................ 10
   2.4 Summary .................................................. 10

3 Interaction between Students and Teaching Assistants ...... 12
   3.1 Background .............................................. 12
   3.2 Initial interviews with participating TAs ................. 14
   3.3 A summary of our observations ........................... 15
      3.3.1 Concepts that many students had difficulty understanding 15
      3.3.2 The nature of interaction between the students and their TAs 16
3.3.3 The approaches that TAs take to help students

4 Focus group - Teaching assistants

4.1 Participants’ background

4.2 Discussion Topics

4.3 Summary of responses

4.3.1 TAs’ understanding of their responsibilities

4.3.2 Discussion about how TAs determine what kind of approach they need to take to help a student

4.3.3 Concepts for which TAs know exactly how to help/ common errors

4.3.4 Examples of when TA was not able to/ found it very hard to help

4.3.5 How much is too much?

4.3.6 Modelling the problem-solving process

4.3.7 Discussion on what the TAs could get out of having a mentor

5 Focus group - Students

5.1 Participants’ background

5.2 Discussion Topics

5.3 Summary of responses

5.3.1 Discussion about when students decide they need the help of a TA

5.3.2 Frequency of seeking help from a TA

5.3.3 Reasons that would stop the students from going to a TA for help

5.3.4 The kind of help expected from TAs

5.3.5 Preference to a particular type of TAs

5.3.6 Differences between learning from a TA and learning from peers or learning in lecture

5.3.7 Discussion about TAs providing motivation
Chapter 1

Introduction

Introductory courses offered in the computer science departments of many universities usually have a very large enrollment as compared to other courses. Virginia Tech is no exception to this case, with over 350 students in CS 1114 – Introduction to software design and over 200 students in CS 2114 – software design and data structures in spring 2016. Both of these courses are mostly intended for students who are minoring or majoring in Computer Science. Students enrolling in these courses usually do not have any programming experience or have very minimal programming experience prior to these courses. For courses with such large enrollments management is usually tough. These courses are often offered in multiple sections.

It is common for introductory computer science courses to adopt a lecture followed by a lab format. Lectures are usually offered by professors/instructors in the department and the lab sections are headed by a pair of teaching assistants consisting of 1 graduate and 1 undergraduate teaching assistant. The lab exercises are usually designed in such a manner so as to enable the students to apply the concepts introduced in lecture. Apart from the lab exercises, students are also expected to complete weekly homework assignments and more elaborate programming assignments once in 2-3 weeks. Most students end up seeking help from their teaching assistants (TAs), either for programming assignments or during lab sections. We wanted to investigate the role of such TAs in teaching these introductory programming courses. More specifically, we wanted to understand what approaches the TAs take to teaching and the rationale behind the same.
1.1 Motivation

Given the amount of enrollments in undergraduate CS degrees across institutions in the United States, including high schools, Computer Science education has clearly received a lot of attention. The Taulbee survey published by Computing Research Association (CRA) points out that the number of undergraduate computing majors has continued to see an upward trend in the last 8 years [21]. Virginia Tech has also seen increase in enrollment numbers as shown below [2].

<table>
<thead>
<tr>
<th></th>
<th>09-10</th>
<th>10-11</th>
<th>11-12</th>
<th>12-13</th>
<th>13-14</th>
<th>14-15</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS Majors Enrolled</td>
<td>304</td>
<td>332</td>
<td>418</td>
<td>420</td>
<td>539</td>
<td>597</td>
</tr>
<tr>
<td>BS Degrees Awarded</td>
<td>101</td>
<td>95</td>
<td>117</td>
<td>128</td>
<td>157</td>
<td>178</td>
</tr>
</tbody>
</table>

Not surprisingly, research in computer science education has kept up with this relentless increase in undergraduate CS enrollment. Interestingly, Computer Science Education research has always had a special love for introductory courses, usually classified as CS1/CS2, as pointed by Dr. David W. Valentine [22]. The reason behind this is because a lot of people have noted that teaching introductory courses is the toughest part of teaching CS curriculum. According to Pears [17], topics in this area of research usually predominantly fall under the following four broad categories:

- **Tools**- Presenting new tools such as those developed to help instruction, improve student understanding through visualizations and animations, enable automated grading and so forth
- **Language choice**- Debate over choice of programming language to be used for instruction
- **Curricula**- Discussions over planning, designing curriculum, including course material design that promote innovative and interesting ways to teach abstract concepts in introductory programming courses.
- **Pedagogy**- Discussions about how teaching and learning happen in these introductory courses.

A fifth and not-so-broad category in CS education research, which gets a little less attention can be thought of as one that aims at promoting diversity in the student group, in the sense of including and retaining women/underrepresented student categories in CS. As an example, Tracy camp et al discusses the enrollment boom and the impact of the boom with respect to how it affects the
underrepresented students [7].

Among all these categories, we are particularly interested in the pedagogy category. Within this broad category most of the research is on pedagogical methods and learning as experienced by students and some on course management.

Being introductory courses, any student intending to major or minor in computer science, is usually required to take these courses, and is required to do so in his/her first year of college education. Apart from such students, others who want to learn basic programming (without an intention to major or minor in CS), also end up choosing these introductory courses. In most institutions, this implies that more often than not, class sizes are usually so large that lectures and labs take place in sections, in this structure students have an opportunity to get more one-on-one time with their respective TAs rather than their course instructors. This kind of individual attention is extremely important when students with little to no prior programming experience are introduced to such new and abstract CS concepts. It would not be an overstatement to say that TAs, especially in these introductory courses, likely have more impact on the student’s learning, than they are given credit for. The quality of the TAs would greatly impact the quality of the course and directly or indirectly also impact the student retention rate and their interest in computer science for their academic/industry careers. So we wanted to investigate the role they play in these courses, by conducting ethnographic studies including observations of their interactions with students, and focus groups meetings, with TAs and with students.

1.2 Research goal

The goal of this thesis is to investigate the role of teaching assistants in introductory programming courses. To do this we observed and investigate the nature of interactions between students and TAs, both during Lab sections and during TA office hours. We focused on understanding the learning difficulties faced by students enrolled in these courses, and we also tried to gauge what the TAs understand their responsibilities to be and the kind of pedagogical techniques they use to teach students, and the reasoning behind using such techniques.
1.3 Background

The 2 introductory courses that were used for this research include CS1114 – Introduction to software design and CS 2114 – Data Structures and software design. While CS 1114 attempts to introduce students to fundamental concepts of programming from an object-oriented perspective, CS 2114 attempts to build on this knowledge to teach more advanced object-oriented software design and algorithm development and analysis. Both these courses use Java as the programming language.

A good majority of the students enrolling in the 2 courses are usually students who intend to be CS majors or CS minors. There are students from other departments who enroll in these introductory courses to learn basic programming. Usually, the students have no prior programming experience with the exception of few students who might have been introduced to programming outside of high school or in their high school AP classes. Both CS 1114 and 2114 take a lecture followed by lab approach, with instructors leading lectures and course teaching assistants leading lab sections. As for tools, CS 1114 uses Sophia.Micro, a pedagogical tool developed for creating microworlds, within a Greenfoot-style environment, by Bowden, for his Ph.D. dissertation [4], from Virginia Tech. CS 2114 makes use of the Eclipse IDE. Both courses make use of Web-CAT[11] for grading programming assignments, which is an advanced automatic grading system that grades based on students’ testing of their own code.

1.4 Approach

At the beginning of this research we were very interested in finding out more about whether a mentoring relationship could emerge between our senior TAs and junior TAs. Our initial interviews with participating TAs, the data from our observation and the survey of existing literature presented the complex nature of interactions between TAs and students in introductory programming courses as a compelling research focus, sidestepping our interest in interactions between TAs.

In order to better understand what TAs do, and how it shapes the learning outcomes of students enrolled in introductory programming course, we decide to conduct some observational studies, interviews and focus groups. The researcher conducted observations of interactions between students and their TAs during lab sections and when the TAs held their office hours. All of the TAs of CS 1114 and CS 21114 and students enrolled who verbally consented to participate were observed
during the course of 2 semesters, fall 2015 and spring 2016. TAs who expressed a willingness to participate in the research, were interviewed individually and were then invite to participate in a focus group meeting. We also conducted a focus group meeting, in 2 parts, with students to understand the students’ perspective on the kind of learning help provided by the TAs. The topics of discussion for this group was intended to mostly mirror that of the focus group conducted with the TAs, so we could identify similarities or differences between the TAs and students take on teaching.

1.5 Organization of this document

Chapter 2 discusses the literature dealing with TAs of introductory programming courses and pedagogical practices. Chapter 3 discusses the data obtained using observational studies at detail. Chapter 4 & 5 summarize the focus group meetings conducted with the TAs and students. The analysis of the findings follow in Chapter 6. We touch upon the implications of those findings and propose possible directions for future work in this area in chapter 7.
Chapter 2

Literature Review

With the continuing increase in class sizes of introductory programming courses, across the country, effective course management has been an issue of concern as well as a topic of research interest, in the CS education community. In this section, we survey the literature about TAs in introductory computer science courses.

2.1 TAs for effective course management

One of the major problems with introductory classes is the wide range of students with varying of experience, making the preparation of material for Lab assignments and their evaluations extremely difficult. Also, in such a large course setting, where enrollment is typically close to a couple of hundred students, it is very possible for students to get a sense of “disconnectedness”, due to lack of scope for teachers to provide individual attention. This also sometimes leads to high attrition rates. Using teaching assistants for leading smaller discussion / Lab sections with fewer students, and distributing grading responsibilities, D.G Kay observes [12], is a common strategy. He also makes a mote of how recruiting Graduate students as TAs is harder in CS, given the plethora of research assistantships and other employment opportunities that they have. To compensate for this difficulty, Kay suggests employing multiple undergraduate TAs, and since employing UTAs is usually less expensive as compared to GTAs, many UTAs could be employed in the place of a GTA without incurring overhead n the department’s expenses. This matches closely with the GTA/UTA startegy employed at Virginia Tech, and in the classes that we observed for this research.
On a similar note, Stuart Reges [18] presents the model that Stanford employed, to allow more students to enroll in introductory computer science courses, improving the quality of education while keeping costs constant. The model involves introducing Undergraduate students as “section leaders” in the place of graduate TAs. These sections usually consisted of 10-12 students. After trying different combinations of graduate/undergraduate students with varying roles and responsibilities, the model evolved to include one graduate head TA per course, a few undergraduate section leaders/graders who worked for credit units and a few experienced undergraduate section leaders/graders who worked for pay. It was observed that the section leading program was not only beneficial to the students in the course but also for the undergraduate section leaders themselves and the entire department.

Eric Roberts [3] notes that using undergraduates as section leaders helps not only in creating an effective learning environment for the students enrolling in the introductory classes, but also for the section leaders themselves. Although they might not have the experience graduate TAs usually have, using undergraduate students as section leaders is more efficient in this set up since the section leaders are able to better relate to the enrolled students, having been in the same situation not long ago. For section leaders, this approach helps them solidify their own understanding of concepts, as they try to explain concepts to the new class.

Adrienne Decker [9] at the University at Buffalo describes the problems with using graduate teaching assistants in introductory programming courses, including the inability of students to connect with their GTAs, and lack of proficiency over the course material, from the GTAs’ side, leading to the evolution of their UTA program. This particular program also involved the UTAs to be involved in the hiring process to recruit replacements for graduating UTAs. In this reflection, Decker reports fewer complaints about TAs in the end-of-semester reviews, a 20% decrease in failure rate, and a higher trend of satisfaction with UTAs, determined by a Likert scale portion of the end-of-semester reviews.

While one the one hand, the use of graduate and undergraduate students as section leaders or TAs as an approach to effective course management in a large class setting was being studied, Dickson [10] discusses the observed benefits of using Undergraduate TAs in small classes in small colleges. The UTAs were observed to have a beneficial effect on students’ moral and attitude. Dickson also explains that undergraduate TAs are able to recall what made the material clear to them, providing good feedback to the professor.
2.2 Training Teaching Assistants

Thinking of teaching as a scholarship, Ernest L. Boyer [5], talks about teaching being a dynamic endeavor that builds bridges between the teacher’s understanding and the student’s learning, using analogies, metaphors and images. He discusses how graduate schools should prioritize teaching and provide their teaching assistants with more help and guidance, considering the needs of those who are being taught seriously.

Jody Nyquist [16], talks about the challenges associated with training TAs to fulfil their responsibilities to their best potential as well as learn from the experience treating it as an internship to enter their academic careers. She discusses a variety of factors including the academic setting, the diversity of the students and their needs, the nature and characteristics of the TAs themselves, their relationship with others in the teaching team and so forth.

Although the works of Boyers and Nyquist’s from as early as in the 1990s, talk about training TAs and the challenges there of in a general sense. Many different universities have their own TA training initiatives for TAs in the computer science department. For example, D.G.Kay [13] from University of California at Irvine, presents a seminar style training program for new CS TAs over a quarter. The seminar course is structured in such a way that the new TAs present lecture material to the group and receive feedback, go over designing assignments and exams and grading programming assignments and exams and designing a course. It is found that TAs who participate in this seminar perform better than their non-participating counterparts on the student evaluations.

Ben Stephenson [20] from the University of Calgary, Canada, discusses a novel approach to training less experienced TAs using peer mentorship. The training program was intended to help graduate TAs to overcome the difficulties faced as a new teaching assistants. In this approach, an exclusive experienced TA as a TA in residence, would organize an optional TA training workshop and also observe the new TAs as they perform their duties and provide them with feedback.

Peg Boyle and Bob Boice [6], put forward a systematic mentoring model, for new graduate Teaching assistants, based off of studies that revealed increase involvement and benefits from being able to reflect on teaching in a group setting, not just for the mentees but also for the mentors. The model included 3 process elements viz. , Planning, Structure and Assessment. Again, without being specific to computer science this study was conducted across all departments, and the mentors and the mentees were not required to be in the same department.
2.3 Teaching methods

As Arto Vihavainen [23], suggests, “Teaching programming is hard”. In his work, he proposes a teaching method based on the principle of learning-by-doing in conjunction with continuous feedback. He claims that learning to program is like learning a craft and for novices learning from people who have already mastered the craft, he suggests using an extreme apprenticeship model, where the Instructors, including the TAs provide as well as receive continuous feedback. Applying this model, they were able to see significant decrease in dropout rates of students.

A lot of work, characterized by ‘explain in plain English’ questions, has been done with regard to investigating the relationship between novice programmers’ ability to write and explain code. Researchers have tried to understand the correlation between a student’s ability to explain a code segments meaning in plain English and his/her success in writing code later and also pedagogical implications of such findings. Examples of such work include [8] and [15].

Michael Kölling [14] proposes a teaching method that includes elements from problem based learning, apprenticeship and use of case studies. They propose a three step approach- Observation, application and design as opposed to the usual approach where students have to start with a design, and then apply concepts taught in lecture, before they can observe the behavior.

Schneider [19], talks about “what needs to be taught” in an introductory programming course without getting into how to teach it. In his work, he lists ten principles that need to be implemented in an introductory programming course. These principles span topics such as problem statement understanding, algorithmic development, programming style, correctness, top-down design, team programming. He concludes that incorporating the ten principles, in the introductory phase itself will likely instill good programming habits in the students and will lead to high quality software production.

2.4 Summary

It can easily be seen that by and large, the work done so far revolves around TAs as being a crucial part about course management, and some about how to train new incoming TAs. There has also been a lot of work on general teaching techniques. Not a lot of work on the actual culture of TAs is available, at least we could not find any ethnographic data on TA’s. We suspect that this could be
because of TAs are generally only looked at as members who help in course management, in terms of leading smaller sections and helping students, and there has not been many studies that effectively measure the impact of TA’s on the students’ learning, especially in introductory CS courses. It is almost like the CS education community thinks of TA’s as being important, but has not thought of them as being that important when it comes to students’ learning in introductory courses. We attempted to capture the culture of TA’s because we believe that understanding what our TAs do can be very beneficial to overall success of introductory courses. The remaining chapters present our findings and the analysis of our findings.
Chapter 3

Interaction between Students and Teaching Assistants

3.1 Background

The research took place over fall 2015 and spring 2016. The fall 2015 session had a total of 269 students enrolled in CS 1114 and 263 students enrolled in CS 2114. The spring 2016 session had 372 students enrolled in CS 1114 and 300+ students in CS 2114. CS 1114 was led by 2 instructors and 6 graduate teaching assistants and 6 undergraduate teaching assistants in the fall and 3 instructors, 7 GTAs and 7 UTAs in the spring. CS 2114 was led by 1 instructor and 5 GTAs and 5 UTAs in the fall and spring. Since the research ran over the course of two semesters, we could not guarantee that the participating TAs were necessarily teaching the same course over both semesters, but at the time of conducting the focus group, all the TAs who participated in the focus group were serving as a TA for one of the 2 considered courses.

The GTAs of both courses were required to hold office hours and grade programming assignments, apart from holding lab sections along with a UTA. None of the TAs were required to take part in the lecture. The courses also each had an online forum on Piazza, where course announcements would be made and students could get help for the assignments, by asking questions to their Instructors and TAs. The lab sections were once every week in both the courses, where the students in a particular lab section would meet with their fellow students and TAs in a classroom, for 2 hours, to finish a prescribed lab exercise, presented at the beginning of the lab, on the course website. The
students were required to physically be present in the lab, to get any credit for their work, but this requirement was changed in spring 2016, for CS 1114 alone. The lab exercise was available to the students at the beginning of the week and it was not mandatory for them to be present at the lab, although they could choose to go to lab where the specific TAs would be available to help them with their questions.

In a lab scenario, some TAs prefer to go to the student’s work area to help the student with his/her questions while some other TAs prefer to have the student walk up to where they are seated. TAs’ office hours, on the other hand, were usually held at what is commonly referred to as the CS lounge. The lounge is open to all CS majors and minors, for working on projects group assignments etc., apart from also serving as place for TAs to hold office hours.

As for the courses, CS 1114 – Introduction to software design holds the following description:

*Fundamental concepts of programming from an object-oriented perspective. Basic software engineering principles and programming skills in a programming language that supports the object-oriented paradigm. Simple data types, control structures, array and string data structures, basic algorithms, testing and debugging. A basic model of the computer as an abstract machine. Modeling and problem-solving skills applicable to programming at this level.*

And CS 21114 – Data structures and software design is described on the VT course website as:

*A programming-intensive exploration of software design concepts and implementation techniques. Builds on knowledge of fundamental object-oriented programming. Advanced object-oriented software design, algorithm development and analysis, and classic data structures. Includes a team-based, semester-long software project.*

During our observations, the researcher, after seeking verbal consent from TAs and students in the lab, tailed the TA and made hand written notes about the interaction, without getting involved in the interaction at any point during the note taking. During office hours, the researcher sought verbal consent from TAs beforehand and from students before their interaction with the TA. We observed close to a 100 interactions, and the researchers notes were then transferred to a computer for further analysis. The observations included interactions involving 8 distinct UTAs and 7 distinct GTAs.
3.2 Initial interviews with participating TAs

The work on this research began halfway through the fall semester of 2015. We started out wanting to find out if an apprenticeship-based mentoring model could emerge between senior and junior TAs of introductory programming courses. 11 TAs (6 graduate TAs, and 5 undergraduate TAs) came forward to participate in this research. We interviewed 5 of them during fall 2015. Following were the questions we asked the 5 participants during their initial interviews.

1. In your opinion, what are generally the difficulties that student face and seek help from a TA?

2. What do you find as challenges in teaching students, especially in an introductory course?

3. What approaches do you take while teaching? If applicable explain with an example?

4. When you are looking at code, with the student, how do you assess their understanding?

5. Do you try to check if the student has really understood the concept? If yes, how would you do it? Provide examples if applicable.

6. Have you ever sought help/guidance from a peer TA with respect to teaching? Provide examples if applicable.

7. Have you ever provided help/guidance to a TA who faced difficulties in maybe explaining a concept to a student? Provide examples if applicable.

8. While seeking or providing help to a peer TA, do you find yourself looking at code? If yes, how does this help? If not, Why not?

Meanwhile we also conducted observations of interactions between TAs and students. Based on our initial round of observations, our research focus shifted more towards TAs’ teaching approaches and their student’s learning outcomes, although our initial interest was to find if interactions between TAs could evolve to follow a mentoring model. To maintain consistency with the research protocol, we interviewed the other 6 participants during spring 2016, asking them the same questions. A lot of the TAs’ responses were consistent with what we observed during our observations. A discussion on our observations are presented below.
3.3 A summary of our observations

We observed the Lab sessions and office hours held by their TAs, after seeking verbal consent from the students in the lab and from those who showed up at the TAs office hours. The researcher was available at the lab or the Lounge where TAs hold their office hours and captured the interaction in writing. The hand written notes were then transferred to a worksheet online, for analysis. About 100 individual interactions were recoded and analyzed. The analysis itself mainly focused towards getting an understanding of the different kind of approaches that the TAs take to teaching a concept, the breadth of concepts that students generally find difficult and ask questions about. Apart from the actual interaction, the data captured the date of the observation, the name of the TA involved, if it was a UTA or GTA, the course and the lab assignment. The findings from the analysis are presented below.

3.3.1 Concepts that many students had difficulty understanding

Based on observing multiple lab sessions and office hours of various TAs, we found that students who are new to programming, typically in CS 1114, find it difficult to grasp concepts including

- object initialization
- using multiple constructors in a class
- method invocation
- parameter passing
- using return values from a method
- loops
- access specifiers

Students in CS 2114, most of whom have taken 1114 or have equivalent minimal programming experience, find it difficult to understand slightly more advanced topics such as recursion, using interfaces and so forth. We found that students in both these courses seek help from TAs for
debugging their code for them. The other common difficulty seems to be that when approaching a problem, while many students have trouble translating their thinking into working code, many others find it hard to even figure out the logical solution to the problem.

### 3.3.2 The nature of interaction between the students and their TAs

Almost always the interactions were initiated by students. This could be attributed to the way that the lab sections or office hours are set up to run. The reason for such initiation, based on what we observed is usually one of the following.

- **Run time errors**
  Students are often expected to write test cases to ensure correctness of their code, and often while executing their tests, they encounter run time errors including the infamous null pointer exception. A null pointer exception in the context of these introductory programming classes, is majorly a manifestation of faulty mental models that the students tend to develop, not understanding the difference between declaring a field and initializing it, or re-declaring a variable with the same name as a field within the local scope of the constructor and so on.

  A **Null pointer exception** could be thought of as the perfect example of the kind of “applied problem” that the TAs help students tackle. The students in these introductory programming classes are taught about what an object is and how to make new objects, during lecture. The actual learning only happens when they get their hands dirty with trying to write their own code, be it for a lab assignment or a programming assignment, when they encounter a null pointer exception. They provide a platform or an opportunity for learning by doing.

- **Not knowing how to proceed**
  In a typical lab session, when the assignment is made available to the students, they go over the instructions provided to them. At that point, a student who has trouble understanding what needs to be accomplished might seek a TA for help. At other times, students seek TAs to help them when they are not sure how to write a particular method, or how to access a field outside of its class and so forth.
Web-CAT feedback related
Web-CAT [11], is the tool that is used for auto grading the lab assignments and programming assignments for both the courses. The tool essentially runs a suit of reference tests against the students code to check for problem coverage and students testing quality as well as mandate java coding conventions. A lot of times, we observed that students after submitting their work to web-CAT, tend to seek a TA for help in understanding and/or fixing the issues as it appears on the feedback provided by the tool.

Testing and debugging
A lot of times, we observed students having difficulty with regards to testing their code. This difficult manifests itself in a variety of ways, as we observed, ranging from issues with setting up a test class, to writing insufficient and non-exhaustive test cases. Since students were not able to appreciate the whole ides of testing, they also had trouble understanding the difference between when web-CAT gave feedback about their testing coverage and problem coverage. The former was about how well the student’s test cases tested their code and the latter was about how the student’s code sufficiently covered all the cases and satisfied the problem requirements, by running it against a suite of reference tests. At times, the students found their own tests failing, and needed help with the debugging process.

Compilation errors
A student who has attempted to write code, might be faced with a compilation error that he/she is unable to fix. This usually happens because of students being unfamiliar with syntax, or not knowing what parameters to pass and so forth.

The following table itemizes the reason for the interaction between the TA and the student and its frequency of it, during our observations. It is important to note that, a lot of times, the reason for initiating the interaction was not the main focus of the interaction itself. For the sake of consistency, we only report counts based on the student’s initiating question and not the actual problem in question.
### Interaction between Students and Teaching Assistants

<table>
<thead>
<tr>
<th>Interaction initiator</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exceptions</td>
<td>27</td>
</tr>
<tr>
<td>Student unsure how to do something</td>
<td>21</td>
</tr>
<tr>
<td>Web-CAT feedback</td>
<td>17</td>
</tr>
<tr>
<td>Students code not behaving as expected</td>
<td>7</td>
</tr>
<tr>
<td>Conceptual understanding</td>
<td>8</td>
</tr>
<tr>
<td>Question about Logic</td>
<td>6</td>
</tr>
<tr>
<td>Compilation error</td>
<td>5</td>
</tr>
<tr>
<td>Writing test cases</td>
<td>4</td>
</tr>
<tr>
<td>Debugging + understanding requirements + questions about efficiency</td>
<td>3</td>
</tr>
</tbody>
</table>

### 3.3.3 The approaches that TAs take to help students

Although we were hoping to see patterns in the data from our observations, such as a pattern tying the kind of question that the student asks to the kind of approach TAs take, or differences in teaching methods adopted by UTAs and GTAs, etc., we did not see any such patterns emerging. The approaches that TAs took to teaching students seemed to be very much ad-hoc and case-by-case. This could also be due to the fact that the researcher was able to capture only the actual interaction. We could only speculate about why a particular TA chose a particular approach to teach a particular student. Despite the fact that we had no means to concretely conclude any reason for the TAs using a certain approach a certain time, we did observe some recurring themes as presented below.

### 3.3.3.1 Diagnosing the students’ problems

TAs generally tend to first go over the student’s code. The TAs also sometimes ask the student to explain what their code does, in an attempt to foreground what the understanding of the student is. This is done usually to understand the student’s thought process. In the case that the student doesn’t know what the problem is, it is usually because the student has wrongly assumed some-
thing about how his/her code will behave. The gap between the students’ understanding of how his/her code behaves and how the code actually behaves becomes visible when the student explains his/her code, the TAs then provide necessary help to bridge the gap. At that point, the TAs might explain the concept themselves, or ask the student to refer a resource. Following is an example.

*S: "comparing string to integer"

*TA: "what are you trying to do here (points at region of code)?"

*S answers

*TA: "what is the scope of this variable?"

*TA explains the return statement, and how it would work inside an if-else construct.

*S had created an "appointment object" with null description inside double nested for loops.

*TA explains how the student has created a new object, instead of adding it into a 2D array.

*TA: "think of it as a slot in your wallet", *TA also refers to previous lab for explaining object assignment.

In the case that the student knows what the problem is, but does not know how to fix it, TAs generally attempt to help the student by prompting or providing some hint about the solution. The hints could also be in the form of leading questions. Most students generally while answering such leading questions, express that they have figured out how to fix the problem.

While sometimes TAs start by hinting at the solution or asking leading questions, sometimes TAs first attempt to gauge the student’s conceptual understanding which may result in providing the student with a quick review of the concept before proceeding any further, and then provide relevant hints.

For questions about documentation and coding conventions usually come up at the end of an assignment, after the student’s code is submitted to web-CAT for grading TAs are generally more willing to simply tell the student how to solve the problem, since it does not involve learning a concept so much as it does inculcating good practices. As an example, following is a note made by the researcher.

*S: "web-CAT is throwing an error"

*TA reads the error and explains that the issue is with naming, student did not use camel case for
method names.

### 3.3.3.2 Techniques for Teaching and Explaining

For explaining the concept, TAs use many teaching methods. A lot of times, TAs took to using the black board or paper and pen, to explain the underlying concept by using the current problem in hand as an example. We also observed TAs using the student’s knowledge based on how they accomplished some task in the previous assignments, to help understand how to solve the present problem. A lot of times TAs also come up with various examples from real life to explain some abstract concept such as inheritance and polymorphism. An example from our researchers notes follows.

*S:* "I am trying to understand inheritance"

TA uses a vehicle->car example. "are all cars vehicles?"

*S: "no"

TA: "are all vehicles bikes?"

*S: "no"

TA: "are all bikes vehicles?"

*S: "yes"

We also observed some of the TAs trying to help the student not only fix the current problem but also learn how to solve a similar problem by themselves, should it occur in the future. As an example of this, during our observations, we saw the TAs explaining to their students what a null pointer exception is, and what would cause it to occur, in a very generic sense, and then have the student try and apply that generic piece of information to his/her code to specifically locate and fix the issue. As another example, we observed TAs who were willing to show the students how to use the debugger tool to debug their own code. Following is an example.

TA: "can you show me the test cases that are failing?" (Student opens the test cases)

*S: "should we try and use the debugger"

TA: "yeah, let us do that. do you want me to help you with that?"

*S: "yeah"

TA explains how to use the debugger on eclipse, he expalins the stepping over function versus step-
As mentioned earlier, we were initially interested in finding out about a mentoring relationship that could emerge between senior and junior TAs, but the complex nature of the interactions between the TAs of an introductory programming course and the students, made understanding the role played by the TAs our immediate research focus. So we followed the observation with 2 focus groups, one each with participating TAs and students, to investigate role of teaching assistants in introductory programming courses.
Chapter 4

Focus group - Teaching assistants

4.1 Participants’ background

The research took place over fall 2015 and spring 2016. In fall 2015, 11 TAs, both undergraduate and graduate, expressed their interest in participating in the research. An initial round of interviews was conducted with the 11 TAs. In spring 2016, 5 out of the 11 TAs who were interviewed initially participated in a focus group meeting. Since the research ran over the course of two semesters, we could not guarantee that the participating TAs were necessarily teaching the same course over both semesters, but at the time of conducting the focus group, all the TAs who participated in the focus group were serving as a TA for one of the 2 considered courses. The participants signed consent forms (in accordance with IRB #15 - 952), allowing us to audio record and utilize the information anonymously. During the focus group meeting, the TAs discussed at length about topics including the approaches they take to teaching, how they perceive their responsibilities as a TA and so forth. The audio recording, was then transcribed, and anonymized before analysis. The following section presents the topics of discussion and summary of responses.

4.2 Discussion Topics

We planned for an open ended discussion based on the following questions as a rough guide. The response of the participants sometimes lead to more follow up questions from the researcher or comments from other participants in the group. Examples were encouraged throughout the discussion for better understanding of the context.
• **Prompt:** What do you think are your responsibilities as TAs of an introductory programming course?

• **Prompt:** When a student approaches you with a particular issue, how do you determine what approach to take to help? (What factors go into determining what or how much you should be explaining?)

• **Prompt:** When do you know that you have helped the student enough? How do you know when the student gets it (getting the idea vs getting the answer)?
  Give an example of a time when you have been very clear in your mind about what concept you were going to teach and how you were going to approach it with the student.
  Can you recall a time when you have tried various approaches to explain a concept/help a student, and failed? What do you think was the reason?

• **Prompt:** Can you remember a time when you tried to help a student by modelling how to solve a problem? (Example: Assuming you are going over code with a student, with the intent of debugging an error, do you think about teaching the practice of debugging itself as opposed to simply explaining what caused the bug and/or how to fix it?)

• **Prompt:** How much is too much and how do you know it is?

• **Prompt:** Have there been times when you have felt good about how you chose to help a student?

• **Prompt:** Have you ever tried to help a student and then thought you could have helped the student in a different manner and he/she would have gotten the concept?

• **Prompt:** What kind of advice/guidance from a mentor, do you think would have helped, when you started out as a TA? Or having the experience now, what advice/guidance would
you have to offer to a mentee in terms of being a TA for an introductory computer science course?

### 4.3 Summary of responses

The questions were supposed to help the TAs to reflect on their teaching experiences, what they do, how they do it and why. The summary of responses follows.

#### 4.3.1 TAs’ understanding of their responsibilities

The responses of the participants when asked about what they thought of as their responsibilities, being TAs of an introductory course, could be categorized into the following broad themes.

- **Teaching in an applied scenario**
  One of the participants who had also served as an instructor for CS 1114 in fall 2015, responded saying that they felt responsible for teaching introductory concepts in an applied situation, such as a lab, where the teaching and learning revolves around a particular assignment. The participant also noted that as a TA they felt responsible to facilitate learning the same material that was presented in lecture in different ways, but revolving around a particular assignment.

- **Help by teaching the process to solving the solution, not by providing the solution**
  Another participant mentioned that they felt responsible for helping the students through their learning obstacles, by showing them how an expert would solve the problem, step by step, without simply giving the solution. On a similar note, another participant said that they felt responsible for helping the students to help themselves, by teaching them how to find and use the right resources.

- **Promoting self-efficacy**
  Two participants responded saying that they feel like it, s their responsibility to make the students feel like they are capable of solving the problems, to build their self-efficacy and
confidence.

- **Inspirit an interest in Computer Science**
  While one of the participants mentioned that because it was an introductory course, they felt like it was their responsibility to make sure that the students didn’t feel turned off by the subject, especially because the students learning style was not catered to. Taking that idea further, two other participants expressed that they felt responsible for instigate an interest in Computer Science in the students. The other participants echoed this opinion and said they attempted to achieve this by sharing their interesting research with their students, by talking to them about all the “cool things” they can do after learning the basics. One of the participants also discussed about how some students feel frustrated when they are not able to solve problems and motivating such students.

- **Providing feedback**
  One participant also mentioned recognizing fair grading and feedback providing as their responsibility. The reason was mentioned to be for the students to feel like their work was being taken seriously.

### 4.3.2 Discussion about how TAs determine what kind of approach they need to take to help a student

During the observation phase, we noticed that the TAs take different approaches while helping students. The nature of this interaction seemed to be very complex, with many variables influencing it. These variables varied from the problem at hand, the student’s understanding of the concept to the time available to the TA, the point in the semester and so forth. So we wanted to hear from the TAs how they determine how to help a student. One participant said that they would gauge if the issue is at a conceptual or at an implementation level, where the student is able to understand the concept but is having difficulty implementing it as code. They said that if it was at a conceptual level, then the approach needed would be to make the student understand the concept better before teaching how to implement it. Another participant said that the gauging happens by asking the student to explain what they are doing, and based on the student’s response, the participant mentioned that they would decide what help to offer and how to offer it. A different TA mentioned that they
would try and teach the debugger tool, early on in the semester, so the students can figure out for themselves, how their code is behaving and what the problem is, especially if the issue was at an implementation level. The same TA also added that if the student seemed to have difficulty understanding the concepts, then the TA would try to use different methods including drawing pictures on a piece of paper, to try and make the student understand the concepts better. The method used, the participant also noted, depended on the content in question.

One other participant mentioned that the students sometime have the mentality to write code on a trial and error basis, where they keep trying different keywords or syntaxes, or throw in different or more lines of code, trying to “hack” their way out of the issue. In order to avoid it, when the TA notices that kind of practice, they said they would take a code walkthrough approach to try and help the student. Another participant who served as a TA for 1114 during the fall semester and 2114 during the spring semester, talked about how knowing the students helps determine what approach to take, for instance, knowing if they generally struggle with the concepts, or if they typically know the concept and just need some help with the nitty gritty details. It was also noted that being able to refer back to what the students learned in the last semester was helpful.

When asked about if the setting, (i.e., helping a student during a lab session or during office hours) influenced what approach the TAs took to teaching, they responded saying that usually in lab sessions, because there is a hard deadline associated with it, they were more willing to provide the student with specific hints to spot the problem and worry less about coding conventions, for instance variable names. They noted that interaction during Office hours would generally be more guiding in nature and aimed at solidifying the concepts rather than providing direct help. It was also noted that, even during office hours, when there are many students trying to seek help from them, typically close to a programming assignment/project deadline, the TAs tend to provide more hints and move on to helping the next student when they think a particular student has enough hints to re-approach the problem. TAs knowing when the student has had enough help/ has understood the concept

On asking the participants how they would figure out if a student has received the required help, or has understood the concept, they responded saying that the students generally are able to signal them if they get the concept. One participant also mentioned that some students try to get the TA to give them the solution upfront and in such a case they would just make a call and encourage the student to try to figure it out themselves with the guidance provided. Additionally, the TA also noted that while doing this, it was also important to let them know that they could always come
back and ask for more help if needed after they have tried. Another participant, on a related note, pointed out that he would almost always just start with providing an overview and get into more details if the student comes back more questions, to encourage them to think for themselves. One participant mentioned that they would have the student explain back to them what they had just taught, or have the student attempt writing the code based on the help while they were still around. Another participant said they would simply ask the student if he/ got it, and just take their word for it.

4.3.3 Concepts for which TAs know exactly how to help / common errors

In an attempt to understand the nature of questions that the TAs usually get asked (including common errors that novice programmers make) and have a tested and tried approach, we asked the TAs to give us examples of times when they knew exactly how to help the student.

Unsurprisingly, Null pointer exception was the first example that came up, and the participants mentioned that they would explain what the error meant and talk a little bit about object initialization and then ask the student about which objects could be null.

The next example that came up was when a student had made calls to the scanner method, `next()`, once inside an `if-condition` and once in an assignment following the `if-condition`, instead of making the call once and storing the result in a temporary variable, before the said `if-condition`, without realizing that is in fact two separate calls returning two different values during run time. The participant who mentioned this example said, that they immediately used a code walkthrough approach to help the student understand what was actually going on.

Another participant mentioned that students usually have a problem understanding how for-each loops work, when they are introduce to it for the very first time and to counter act this, they said that they would make use of a piece of paper, to write down in natural language a phrase with two blanks for the student to remember and fill out as required, such as “for each __________ in collection __________ “.

One participant pointed out that some students, during the early part of the semester, are confused about when to use `if` conditions and when to use `while` loops. The participant noted that the approach that usually helped was using a real life scenario to bring out the difference between the two.
Other participants mentioned scope issues and using new syntax as some common issues that the students seek help for, and said that asking leading questions or trying to relate to a previously known concept to the student helped in such cases.

4.3.4 Examples of when TA was not able to / found it very hard to help

We asked the participants to recall if there was a time when it was not very obvious what approach they needed to take to teach a student a particular concept. One participant started off the discussion by responding that generally it is difficult to explain to the student about the errors thrown by the reference tests on web-CAT. Another participant, spinning off of that, mentioned how most students in such introductory courses are unfamiliar with Boolean algebra, which makes it particularly hard for them to understand testing of all possibilities while writing if conditions. The participant noted that in such a case, he would try a couple of times to explain Boolean logic using heads and tails example or on and off examples, but if the student still had trouble understanding, the participant noted that they would tell the student how to solve the issue and hope that they understand the Boolean logic based on seeing how the solution worked.

Another participant recalled that teaching the students an abstract concept such as an interface was particularly hard, in spite of trying approaches like drawing analogies, or drawing diagrams and pictures. This opinion led the group to get into discussing how the introductory courses tend to throw a lot at the students, in a little time. As mentioned earlier, the participant who had also served as an instructor shared with the group the reason behind that. The reason was for students to feel like they have a lot of power, being introduced to so many different things, that they can use, to get them excited about it, postponing the dealing with details for until later courses. The participants then discussed how they felt about it and one participant mentioned how they felt like this methodology did the exact opposite of what it claimed to do.

Another participant expressed that motivating reason behind Test driven development was difficult to get across to the students. Discussions followed and participants seemed to all feel like students perceived the activity of testing as something that needed to be done with without appreciating the importance of it.

One participant stated that making students understand references was also very hard. The students’ unfamiliarity with how things are organized in memory was understood as the reason for
this difficulty.

4.3.5 How much is too much?

After discussing about the concepts that the participants felt as particularly hard, all seemed to have one thing in common – all of the said issues were mainly because students were unfamiliar with related concepts such as Boolean algebra, or memory organization. In an attempt not to overwhelm the students, sometimes TAs are forced to provide the solution upfront or just briefly introduce the topic to the student. On asking the participants about how much they thought was too much and how they figured that, we heard the following.

One of the participants mentioned that they generally try to do a very brief summary of the topic, and assuring the student that the topic will be covered in more detail later, if the student seems to be overwhelmed. The participant also mentioned not knowing sure if the topic was covered enough or if the student understood any of it or even how much the students are supposed to know

Another participating TA mentioned that they would take a step back when they catches themself trying to explaining the memory stack. One other participant mentioned how sometimes they wanted to explain why something would work a certain way, in a certain programming language and not in others, but then realizes that, that wouldn’t mean anything to them. The same participant also mentioned, acknowledging that it might be biased, that to gauge if they were explaining too much by asking the student if he/she is a going to be a CS major or minor, because students who take these courses simply to learn basic programming typically don’t really need to know such details.

4.3.6 Modelling the problem-solving process

We saw during the observation process a kind of process where the TAs would show the student how to go about solving a problem or debug an error. To hear firsthand from the TAs, we asked the group if they made an effort to consciously model the process to the student, like an expert would do for a novice in an apprenticeship-based learning model.

One participant mentioned that by having the student do a step by step debugging process, starting, from reading and understanding the error message to tracing back to the problem, under her guid-
ance, was her way of modelling the debugging process, so that over time the student learns to do exactly that on their own, when they encounter an error in the future. Another participant echoed a similar process.

One participant mentioned as a TA, they might have been modelling the process without consciously thinking about it. They also mentioned how that might be bad because sometimes the students could pick up the informal practices that they adopt while trying to help them. The participant noted that sometimes them questioning about what the students code was doing was merely an attempt at understanding the students thought process and sometimes also mistaken to be a leading question to the student.

The participants also mentioned drawing pictures for the students, or asking the student to explain the task at hand in plain English as a form of modelling the problem solving process. Another participant reflected on the experience they typically has with 1114 students more where when asked to explain the code, the student would still use the keywords in the code as opposed to explaining in plain English. Laurie Murphy [15], probe into exactly this phenomenon where they note that having students answer explaining questions helps brings out faulty mental models that the students might be building.

4.3.7 Discussion on what the TAs could get out of having a mentor

As a sideline, we asked our participants what they wish a mentor would have told them, with respect to being a TA for an introductory course, if they had one or what advice they would give to a new TA, if they could be a mentor. One of the participants mentioned that it would have been nice to have more context about the curriculum at Virginia Tech. Another participant mentioned that it would be good advice for new TAs of an introductory course to remember that they are trying to help the students through and not weed them out, and the kind of impact they have on their own students. Another participant mentioned that trying to remember when the TAs were students themselves, would help them relate to their students better. Another advice that the participants thought that new TAs could use was to assume that students don’t fully know the concept, and to probe the students understanding before attempting to help with any coding.
Chapter 5

Focus group - Students

5.1 Participants’ background

We had a total of 4 students participate in a focus group that was held in 2 separate sessions, due to scheduling difficulties. All participants were enrolled in either of the two courses during spring 2016, when the focus group was conducted. 3 out of 4 participants who were enrolled in CS 2114 had also taken CS 1114 in fall 2015. The participants signed consent forms (in accordance with IRB #15 - 952), confirming that they were all over 18 years of age and allowing us to audio record and utilize the information anonymously. They also filled out a background questionnaire which had questions about their gender, year of study, major and some scenario based questions regarding their preference of TA for seeking help, the responses are presented below.

1. Student background information
   All 4 participants were male and over 18 years of Age. 3 students had Computer Science as their minor. Their majors were Computational Modelling and Data analytics, Applied and Discrete Mathematics, and Electrical Engineering. One participant was a Computer Science Major. 3 out of 4 participants were in their junior year, and one participant was in his sophomore year at the time of the focus group meeting.

2. Responses to questions about seeking help from TAs
   We asked the participants if they would prefer seeking help from a TA, seeking help from peers or solve a problem by themselves. 2 participants responded saying that they preferred seeking help from TAs. 1 participant each referred solving a problem on their own and seeking help from peers.
As a response to the question about how comfortable they were asking for help from a TA during lab sessions, 3 out of 4 participants said they were very comfortable, and 1 participant responded that they were somewhat comfortable.

We asked the participants to indicate what mode of contact they use to get help from a TA. 2 out of 4 responded saying email was their usual mode of contact. 1 participant indicated face to face during office hours or through email, as their usual mode of contact. 1 participant seemed to have misunderstood the question and responded with “not directly seeking for an answer but an approach”.

On asking about the frequency of their visit to TA office hours, the participants responded with “not frequently”, “twice a month”, “very often” and “rarely”.

For the question about what reason mostly solicits help from TAs, among homework assignments, Programming assignments, lab exercises and general concepts taught in lecture, all 4 participants chose lab exercise, and 3 out of 4 participants also chose programming assignments. None of the participants chose general concepts taught in lecture or homework assignment to be the reason for seeking help from a TA.

We also wanted to find out if for any reason, students preferred to get help from UTAs or GTAs. As a response to this question, Participant 1 pointed out that UTAs help was needed more for logistical issues like grades, and GTAs help was more needed at the programming level. Participant 2 preferred GTAs, mentioning that they seemed to know more. Participant 3 mentioned that UTAs were preferred since they were more experienced. Participant 4 had no preference.

The questionnaire had the following scenario-based question, and the participants were asked to indicate their preference with a reasoning.

Consider the following Scenario:
Your program is throwing an error that you have not encountered before. You do not know how to solve the issue. You decide to visit a TA for help. You have 4 options.
TA 1: This TA would quickly help you fix the error, simply by looking at your code and telling you exactly what to change in your code to fix the error.
TA 2: This TA would go over your code with you to explain why the error occurred and would ask you to fix the cause for the error.

TA 3: This TA would explain the error and the typical cases in which this error message would be thrown, as hints for you to understand the issue.

TA 4: This TA would ask you to go over the code and make you reflect on what the state of the system is at each step of the code and eventually lead you to the erroneous state, so you can fix the issue.

Which of these TAs would you prefer to seek help from? Please explain why.

The 4 choices were intended to range from a TA 1 who would provide quick and direct help, like a short term specific help, to TA 4 who would provide help on a very generic level, the kind of help that will help the student in the long run. Participant 1 preferred TA 3, with the reason that the kind of help from TA 3 was generic enough to avoid future errors of the same type unlike TA 4, whose help could be efficiently replaced by googling the error message and solving the problem themselves. Participant 2 marked their preference as either TA 2 or TA 4, stating that those TAs, as their approaches were balanced in the sense of guiding the student towards the answer as well as making them understand the process of doing it. Participant 3 preferred TA 2, and mentioned that that kind of help would be direct and save them more time. Participant 4 preferred TA 1 stating that the providing the fix would be sufficient for understanding the problem, and that the student would ask more questions if that is not the case.

5.2 Discussion Topics

Similar to the Focus group for teaching assistants, we wanted to conduct an open ended discussion, based on the following prompts guiding the discussion. These prompts were designed, after the focus group for TAs was conducted, with an intention to understand the students’ take on seeking help from a TA, to be able to draw similarities and differences between the students’ views and the TAs’ views. We asked them about the differences between getting help from a TA in different situations, such as during a lab session and during office hours. We also asked the participants about how seeking help from a TA differs from other modes of learning, to understand the students’ firsthand about the teaching role that the TAs take on.
• **Prompt:** When do you go to a TA for help? At what point?

• **Prompt:** How often do you seek the TA for help?

• **Prompt:** Have you ever stopped yourself from going to a TA for help? If yes, can you describe why?

• **Prompt:** What is the kind of help that you expect from a TA? Does it depend on any factors? (Is it always the same expectation or does it depend on factors?)

• **Prompt:** Do you prefer to go to a particular TA/set of TAs for help? And why?

• **Prompt:** How is getting help/learning from a TA different from learning in lecture/ getting help from peers (consider help from online discussions on forums, do you find it more effective? why? why not?)?

• **Prompt:** Is there any difference between talking to a TA during office hours and talking to a TA during lab sections?

• **Prompt:** Have you ever had an experience when you did not get the help that you wanted from a TA?

• **Prompt:** Have you ever felt encouraged/motivated after receiving help from a TA?
5.3 Summary of responses

5.3.1 Discussion about when students decide they need the help of a TA

On asking the participants at what point in the process, they would decide that they need the help of a TA, one participant mentioned that he would decide to go to the TA for help after trying to figure the problem out for about an hour or so.

Another participant mentioned that depending on if the problem was logic related or syntax related, he would decide how long to work on the problem by himself before approaching a TA. He mentioned that if it was a syntax based question, and if he wasn’t able to figure it out himself in less than a minute, he would ask the TA immediately, whereas if it was logic based, he said that he would try for as long as an hour and then ask the TA. The reason he said was asking the TAs about syntax was more efficient than trying to solve it himself. Asking the TAs about a logical problem as opposed to trying to solve it himself, he mentioned was less efficient because of the tedious process of having the TA caught up to speed about his problem solving process and the uncertainty that the TA will be able to spot the error after all that tedious process.

On being asked about a time, when the student was unclear about how to start doing something, another participant mentioned that situation arising often during lab sessions and that he would ask the TA in the lab right away, especially because there is a specified amount of time within which the students are expected to complete the lab assignment. When not in lab, if such a situation arises, the participant mentioned that he would try a couple of different ways to approach the problem, before seeking the TAs help.

5.3.2 Frequency of seeking help from a TA

We wanted to get a sense of how often the participants interact with the TAs for getting help with their problems. The common theme that emerged was that the participants approached the TAs way more often in labs, and mainly because of the associated time limit.

One participant mentioned that he would seek help from a TA for a project assignment, during
their office hours, only when the programming assignment makes use of a lot of concepts that he is unfamiliar with. Another participant mentioned that going to a TA for help would be his last resort. He added that, after figuring out most of the solution, he might go to the TA for help with regard to deduction of points on web-CAT and occasionally, to ask the TAs opinion on improving his code quality. Another participant echoed a similar opinion and additionally noted that as the end of the semester approached, the number of visits to TA increased due to busy schedules and reduced time available for devoting to projects.

The participants who were enrolled in CS 2114 specifically spoke about how it was particularly demanding to be able to solve the lab exercises in 2 hours, especially because they are building off of whatever blocks of code they are provided with and not writing their own from scratch. This in their opinion meant that the solution expected out of them was pretty specific and it meant that the whole process could be considered as try different ways to approach the problem until the TA tells them what the right approach would be. On probing further, the participants did mention that trying all the other incorrect approaches and failing did help them better understand why the approach that the TA would recommend, was correct. But the take away from this discussion was that within the 2 hour limit, the nature of the problem almost always involved seeking the TAs help in lab after all the failed attempts by the student, to get to the solution and did not give much room for “try and fail till you find the solution yourself”.

5.3.3 Reasons that would stop the students from going to a TA for help

One participant mentioned that he would try to avoid going to a TA for help, to feel like he is capable of solving a problem without the help of a TA, all by himself. Another participant mentioned that the only reason that would stop him from going to a TA was the inconvenience of going to office hours. He mentioned that if the TA was available where he was, like in lab sessions, he would want to utilize the help that they can offer him. But for visiting during office hours, he noted, was not worth it, unless he knew upfront that he would need a lot of help due to the unfamiliarity of concepts involved in solving the project.

One participant responded that he would choose not to go see a TA for help when it is close to a project deadline. He said that during such “crunch time” TAs tend to have too many students visiting them, all of them trying to get help at the same time. In such a scenario, TAs tend to provide sparse hints because of the high demand and because they are trying to help the all. The wait time
to get to talk to the TA along with the amount of time or the kind of help the TA is able to provide during such a busy hour, he said would deter him from going in the first place.

The same participant also said that a lot of TAs being international students, whose native language is not English, tend to have thick accents and that adds a language barrier on top of the technical barrier. In such cases, the participant said he would prefer to go to a different TA who can communicate better, if there is one.

Another participant echoed the “avoid going to the TAs for help during crunch time” calling it a tradeoff, deciding between going to the office hours and getting the right answer from the TA without being able to comprehend why or trying to figure out the solution yourself if he can afford the time. There was also a mention of how there was no way of figuring out an error, that does not occur so much from a lack of reasoning or application but because of some specific technicality, without going to the TA. For instance, an error thrown because web-CAT expects the output string to be in a specific format and the student’s output string had an additional space character.

This discussion slowly digressed to difficulties or issues that the students face while trying to get a TAs help on a project, during their office hours. One participant mentioned that with problems that may have different correct solutions, it is difficult for a TA to be able to go over the student’s code and understand the thought process and then help debug the issue. At that point, sometimes the TAs are forced to suggest a completely different approach to circumvent the issue, which would imply that the student discard most of his current work.

Other issues that came up during the discussion included the informal setting in which the office hours are conducted, the lack of a proper queue system as some students tend to be more aggressively trying to get a turn to talk to the TA. The participants also mentioned that there have been times when they have not been able to find the TA that they went looking for, given that students can approach any of the course TAs, and not just their own lab TAs.

5.3.4 The kind of help expected from TAs

One participant mentioned that when there are too many students trying to get help from a TA, then he would like to be told at least what he should be doing, maybe give an example and verbally
state what the ode should look like. But in an ideal scenario, the participant mentioned that they expect TAs to give them more help than they can get for themselves by googling. Specifically, he said he would expect the TA to be able to look over his code and see where the issue is, and even if not tell him directly, he would want the TA to be able to guide him in the right direction so he can go figure it out himself, from that point forward.

Another participant shared the same view, but also recognized that he cannot possibly expect the TAs to be able to spot the errors given that there are so many ways one could devise a solution to a particular problem opening too many possibilities for errors to occur. Both participants also mentioned that as the time to the submission deadline decreases and frustration of not being able to figure out the issue or the fix to it increases, the tendency to want the specific answer from the TA increases. Participants also admitted to procrastinating as the reason why there are so many students during TA office hours close to a deadline.

One participant mentioned that the kind of help he would expect from the TA would vary with the kind of problem he is trying to get help for. He said that if he was going to approach the TA for getting help with a conceptual question, then he would expect them to help them get a deeper understanding of the concept, whereas if it was a specific question on a project, he would expect the help to also be specific.

One participant mentioned that he would always prefer the direct answer. He noted that if he doesn’t understand the specific help given to him, he would always ask for more help, and that he would personally prefer to be the one to drive the learning process forward, if he wishes to know more.

5.3.5 Preference to a particular type of TAs

One participant was able to recall a specific TA and noted that he preferred going to that TA for most of his questions. Although the TA did not provide him the kind of specific help he expected, instead chose to guide him how to approach similar problems, he noted that when he finally solved the problem, he felt a sense of accomplishment. Another participant mentioned that he has not had a lot of experience going to TAs to have a preference, but he did mention that he would always prefer to go to a TA who would give him specific and direct answers.
Another participant mentioned that he would go to a TA who he think communicates better and is likely to know the problem faster, someone with more experience being a TA for the course. Another participant mentioned that he had developed a good working relationship with the TA in charge for his lab, and for the same reason he said he would prefer to go to her office hours for doubts and questions over a random TA's. Another participant echoed this opinion and added that because he was not very comfortable having to open his work to criticism from a different TA.

5.3.6 Differences between learning from a TA and learning from peers or learning in lecture

One participant mentioned that learning in lecture is awful, because there are too many students with different levels of coding experience, and the lectures generally tend to keep it slow paced, so everybody gets to keep up. He mentioned that with going to a TA for help, he could ask them for help with very specific things.

Another participant responded saying that in lecture, a lot of concepts are introduced and sometimes when he faces a difficulty trying to understand the concepts, he would go to the TA for it, before the issue cascades to the next lecture. One participant agreed that the lecture tends to give them the basics and they are expected to learn more on their own, and the TAs act like the recitation that comes after the lecture. He said that asking a TA about a specific example, that might not be relevant to the rest of the class, would be better than taking up that amount of class time, and especially so because one question’s answer generally spawns a couple of more follow up questions.

Another participant mentioned that during lecture, he gets to see a snippet of code and how it works and what it does, but not actually using it. With a TA, he said, it is where the actual learning-by-doing happens, and even if the TA was to give him the answer, he would learn more by it because he would have already gotten his hands dirty and tried failed before.

When asked about learning from peers, the responses included comments about how much a peer can help might be limited by the assignment, and how the kind of learning from a peer would largely depend on how good the peer is with respect to the concept. One participant said that with peers, the kind of help is usually very solution oriented and almost never conceptual. He mentioned
that it was a lot easier to ask a friend/peer for help to try and finish the assignment.

On asking about the kind of learning that happens over an online class forum, one participant mentioned that asking a question on the Piazza, almost always guarantees a response and is convenient to use from wherever but there is also a delay, which he called “unsatisfying” involved with getting a follow-up question or two. This he mentioned was a difference between talking to a TA face to face and waiting after posting on Piazza.

Another participant mentioned that Piazza is generally never used for general learning, and that a student would generally be looking at other posts only to see if a different student has faced a similar issue and posted on Piazza, during the same window of time. As far as the learning itself is concerned, to be able to see some others have approached a similar problem, for instance, the way other students have set up their classes, even all the actual code is not posted, the participants said was useful. One participant said that working with a hint that was given on Piazza, is much more convenient because it can be referred to and re-read again many times at a later point in time. He also noted that people are generally more methodical with respect to asking questions on Piazza because it is all documented, and the responses tend to be more composed, keeping in thought the best way to teach somebody. With all these alternatives, two participants mentioned that talking to a TA would be the ideal way for them to get help, provided they are able to have enough time with the TA.

5.3.7 Discussion about TAs providing motivation

One participant when asked if he felt motivated or encouraged, after receiving help from a TA responded saying that he did not recall such an incident. Another participant called himself a highly motivated student and said that he has never particularly needed an explicit motivation. He also mentioned that he would think of triggering his interest in the field more and showing the bigger picture of why he is learning these basic concepts, would be something. Another participant mentioned that he likes it when the TAs recognize that he has attempted to write neat looking efficient code, not just functional code. He said that would be motivating him further to keep at it. He also added that it is encouraging when the TAs can relate to the kinds of trouble students face, for example with trying to test all possibilities of an if block to satisfy web-CAT, if the TA understands and is able to relate to it, he said that that would be encouraging to know that extensive testing is in fact difficult and that he is not the only one finding it hard.
All participants agreed that they have never encountered a scenario where they felt discouraged by a TA. One participant mentioned that there was usually no time for the TA to provide the student about his progress, face-to-face. The participant also felt that the feedback they provide on the project submissions on web-CAT is also problem specific. One of the participants added to it saying he would prefer to hear a positive comment about his work in person over reading it on web-CAT.
Chapter 6

An analysis of our findings

Before we begin to present our analysis from our findings, we would like to acknowledge the limiting aspects of the approaches that we followed to do this research.

- The data from our observations, due to the nature of the process, provides no concrete evidence as to why the TAs did what they did. We are only able to analyze and present what we observed.

- Due to scheduling difficulties, we held our focus group with the students in a fractured manner. Although we used the same prompts with both participants in both sessions, we were unable to capture how the participants felt about certain topics that came up as a result of a particular discussion in the other session.

The following sections attempt to present an analysis of our findings that we believe are particularly worth discussing, based on our Observations and the responses from our focus group participants.
6.1 Implications

6.1.1 The Gap between TAs’ intentions and goals and students’ expectations

In our observations, we saw that the TAs seldom provide any direct help to the student, instead push the student in the right direction by providing hints, or by making the student explicitly process his/her understanding of a particular concept by engaging in the act of asking leading questions. We speculate, based on our observations, that the TAs were in fact modelling the problem-solving process for the student to observe and imitate. To confirm the intent of this behavior, we asked the TAs who participated in our focus group, and they responded in the affirmative. One of our TAs said

“that is exactly what I am doing when I ask them to hover, I will be like “hold your mouse over the error for me so that it will pop up with whatever exception or whatever the problem is” and I will ask them like “what is that, why is that there “ kinda thing, So I am trying to lead the, so that next time they have it, they can look at the error message and be like “I have had this before” , that’s why I teach them the debugger too, because I’ll be like “oh we have a null pointer exception, drop a breakpoint on this line for the test you just failed, now go into the debugger and hold your mouse over the objects, and that one’s null,why is it null ?” and they can go and find where it wasn’t assigned or it was never initialized so hopefully that will be modelling for them what the debugging process looks like , in the future, “Go look at the debugger, what’s the problem?”."

While the evidence leads us to believe that the TAs are almost always intending to provide guidance that is going to help the student help themselves in the long run. On the other hand, in our observations, we saw a tendency in the students to be very concerned with the immediate problem at hand, and seeing the TA as someone who holds the solution to that immediate problem. This tendency was also evident when we spoke to the 4 students who participated in our focus groups. For example, on asking about what kind of help he would expect from a TA, one student responded saying

“in the ideal scenario where every kid had a GTA, they are really involved where the read the code
and tell you what you are doing wrong, personally I prefer them, if you are going to a GTA, they should give you more help than you can get yourself by googling. There is no point going if they are just going to tell you like “oh this error is usually because of input error” ut I googled the error yeah, I know it’s for an input error, that doesn’t help me at all. So it has to be more involved, with them looking at your code and see where the error is, and maybe not tell you directly but tell you to look over a portion of your code, and then you can go out and look for it and if you know you have narrowed it down to one class, then you personally can look through it and figure out the error on your own. Just telling you what the error is, that’s no help there.”

So the immediate questions that arise are about this visible gap in between what the TAs aim to do and what the students are expecting from their students. Is this gap healthy? Is it at all possible to bridge this gap? Do we need to bridge this gap? Is there any way we could resort to, in order to increase the perceived value of the dialog between the TAs and their students? Although we do not have answers to these questions, we see that thinking about this gap could be very useful while structuring the courses and involving graduate and undergraduate TAs as members of the teaching community. This gap could mean that some students avoid going to TAs for getting help and are actually losing out on the resource that is available to them. It could be the case that this gap is in fact healthy, in a sense that it deters the students from thinking of TAs as an alternative resource to get the solution from. In any case, being mindful of this gap we believe will help the CS education community when it comes to structuring and designing an introductory programming course.

6.1.2 Are the time constrained exercise taking away from students’ learning?

What we observed about interactions during office hours, especially right before a particular programming assignment/project was due, was the increased number of students who were trying to get help from the TAs. Our TA participants also confirmed that time is a huge factor that influences the teaching approach that they choose to take with the students. Especially, during lab sections, the students are typically working on a particular exercise, that is designed with an intention to allow the students to practically apply the concepts introduced during lecture. But because, there is a hard deadline associated with the lab, which is counted towards the students grade in the course, it seems to be the case that majority of the time, there is a focus shift happening from learning the
Due to resource constraints, in spring 2016, CS 1114 labs underwent a change. The students were no longer required to be physically present in the labs. The exercise was available to them during the start of the week, and the students had up to their lab time to submit their solutions. Specific TAs were present in the lab sections to help the students who showed up to lab with their questions. The idea was that the students had a chance to at least go over the lab beforehand and could work on it ahead of time, and receive help from TAs during the scheduled two hours.

In CS 2114, the students are usually provided with a pre-lab, that they are expected to complete before lab time, to prepare them for the lab exercise. But our findings revealed that a lot of times, students have difficulty with completing the exercise within the stipulated 2 hours, and most often are forced to seek TA help, because they cannot afford to spend more time trying to figure out the solutions themselves. During the focus group meeting, one of our participating TAs also mentioned that they tend to not press over things like good coding practices during lab sessions, because of the associated time constraint. Based on our findings, we wonder if the time constrained activities take away from the student’s learning.

6.1.3 Providing our new TAs with collated pieces of advice, to help them ease into the process

As we mentioned earlier, the department of computer science at Virginia Tech, currently does not hold any formal training programs for its new TAs. The new TAs are on their own and eventually figure out for themselves about how to do their job. But based on our observations we are able to clearly see that the TAs face a number of situations where they have to make decisions, that directly or indirectly impact the learning of the student’s. These decision points include and are not limited to the following:

- Deciding what kind of help the student needs – This could be based on if the students problem is at a conceptual level or at an implementational level or based on external factors such as time that the TA can spend with a particular student, when there are too many students waiting in queue, which is again influenced by the time left before the submission is due.

- Deciding how to provide that help – Decision making in this category would involve deciding
from among a variety of pedagogical approaches.

- Deciding **how much** help to provide Some students have the tendency to try and keep asking the TA for help repeatedly, to get the TA to give them as much of the solution as possible. The TAs at that point need to determine where to draw the line and push such students away and have them try to figure it out on their own before asking for more help.

On the other hand, sometimes TAs also are faced with a situation where they catch themselves explaining a concept that the student is unfamiliar with. Such concepts maybe related concepts that the students are yet to be introduced to or some low level details that might help the student understand the current concept better. In such scenarios, the TAs are forced to make a decision on how much they need to explain. At that point the immediate concern is the student feeling overwhelmed with new and unfamiliar, possibly more complex concepts. One of our TAs’ Focus group participant said

“and some of them have, sometimes, even to gauge of this might be bad, this might be like biased, but to gauge how much I should tell them, I ask,” are you a major or a minor “, or are you even going like far in this, cause maybe you don’t care and you are not taking anything really after this, I mean some people are their main focus is they are mechanical engineers and they are taking this to get a basic understanding of programming, but they don’t wanna know like the guts of all of it so but then if they say like “I am a major” then I am like “ Okay, you will need this later” or “you need to know this” I don’t know. Maybe that’s biasing my students and preventing some from learning..”

Clearly, our TAs are faced with many situations where they have to make a call about what they should be doing next, and there are many different factors that influence such decisions. Whethere the student was going to major or minor in Computer Science, also apparently one of those factors. Our new TAs need explicit guidance with regard to handling such situations, especially because at that point they are making a decision regarding what to teach or how to teach it.

When we asked our TAs during the focus group meeting about what would have been good advice to them when they started out as TAs, they had different suggestions. We also had them discussing
about some problems that are particularly hard for them to teach students. Although we did not find any evidence for when our TAs were not able to help the student at all, it is very possible that there might arise a situation where the TAs may not be sure about their abilities to provide help to students. All these findings makes us want to think about if we could help our new TAs by maybe providing guidance and advice about such situations. If we were to think of a mentoring approach, the questions we should be asking ourselves include what would constitute as help and how would we be providing that help.
Chapter 7

Conclusions

Our interest in investigating the role of TAs in introductory programming courses was set off by observing the TAs interact with their students, during our initial observations. At Virginia Tech, new TAs in the computer science department are typically not provided with any formal training about teaching methods or strategies and are pretty much set out to learn on the fly. Sometimes, instructors of introductory programming courses choose to hold organizational meetings, typically once a week, with all the TAs to keep abreast with course progress. These meetings usually are aimed at making sure that there is consistency among all the teaching members involved. Usual discussion points include going over a project grading rubric, agreeing on timelines, upcoming course events etc.

Our observations revealed that our TAs used many different teaching methods and many different techniques, at different times. They addressed the class as a whole when they saw that a lot of students were facing difficulty trying to achieve something, they drew diagrams on paper, they came up with many different examples, and they asked questions till the student figured out how to solve the problem themselves. The immediate thought we had was how do the TAs decide what approach to take? How do they know when it works and when it doesn’t? What do they think are their responsibilities? Do the students see them the same way that they see themselves? What do the students really want from their TAs? To answer these questions, we spoke to the TAs and students themselves.

What we heard from our participants was remarkable. All the TAs who participated in the focus group meeting had one thing in common: they seemed to be thinking of themselves as not someone
who was there for the student to solve their errors and fix their bugs, but actively help them learn and engage in the process. Almost all the conversations that we observed showed some effort on behalf of our TAs to actively explain concepts, looking at the students’ problems as opportunities for them to teach the student about a new concept.

The students who participated in our focus groups, helped throw some light on how they see the TAs as a useful resource, and how they are not able to get enough out of the resource when sometimes the demand spikes.

Computer science education related conferences may have seen many pedagogical tools that can help manage a course, provide a platform for enabling discussions online, automate the grading and feedback process etc. But teaching and learning in CS is so intricate and complex, we believe that any tool can go so far. Especially in introductory courses, TAs play a very crucial role in providing the kind of individual attention to students that they cannot get in a lecture, with usually 100+ students.

While we were observing students, we began to wonder how learning how to program sets them in the path of computational thinking, which essentially means to be able to problem-solve using algorithmic thinking, abstracted models, and being able to generalize this process to many different problems. The courses themselves seemed to be set up with an intention of not only introducing the students to various language specific – constructs, but to encourage a way of computational thinking. This is as far as the intent goes. When applied in a concrete scenario, with a combination of active participants in the process: the students, the TAs, the tools being used, the focus can easily be misplaced. It happens to be the case that the tools and students frustrations can cause this focus shift, and the TAs are responsible for not letting it happen.

A lot of times, during our observation, we saw students initiating conversations with their TAs after submitting their work to web-CAT. Most times, the TAs were able to help the student understand why certain deductions were made and the students went back and right their mistakes. But some other times, due to external pressures such as lack of time, the students were either not able to get enough explanation as to why the deductions were made and why such checks are present in the tool in the first place, or that they were already too frustrated to appreciate the importance of good coding practices/ extensive testing, that they start doing something simply to stop web-CAT
from complaining or deducting points. The student is no longer thinking of learning to test extensively, or follow coding conventions as important but only as something that needs to be done “to keep web-CAT happy”. As a different face to the same problem, some students, for example, put in empty or meaningless Javadoc’s, in an attempt to simply bypass web-CAT’s checks. At this point, TAs become indispensable, to motivate the underlying reasons, and to effectively bridge the gap there is from learning to program to being able to problem-solve, in a more one-on-one setting.

Also the students themselves generally do not have an idea as to why they are learning what they are learning in these introductory courses, and what comes after. We ran our 2 focus groups with 5 TAs and 4 students. Although this might not be a good number to generalize the findings, we still cannot discount the observed trend for nothing. All our TAs consistently thought of inspiring an interest in computer science in the students as one of their responsibility. We believe that this understanding helps them keep the focus on setting the students in the path of computational thinking, without being bogged down by learning to code in a particular programming language.

### 7.1 Future Work

We believe that the data that we have collected from conducting observations and focus groups can be used as an evidence, for the kind of intuition that the TAs possess when it comes to teaching introductory concepts to students. This can be leveraged in one of three ways described below.

- **To conduct a contrastive study**
  To evaluate, maybe quantitatively, the learning outcomes of students who interact with TAs who each adopt different strategy. This kind of a study, may help uncover useful insights about the types of students, what teaching techniques cater to their learning styles, and how each group of students achieve learning.

- **Training of TAs**
  The data and findings of this study can be used to inform an administrative change in the department, which may include introducing a new role in the course team, a trainer whose work would be to effectively train and help the TAs and act as a liaison between the instructors, department and the TAs. This model might help take some of the administrative load off of the instructors’ hands while also establishing a continuous channel for feedback to and
from the TAs.

- **Introducing a mentoring model**
  Our findings suggest that our TAs tend to broadly see the kinds of difficulties that are students face as either being predominantly at a conceptual level or at an implementational level. We also heard from our student participants that the time-limited labs shift away from the goal of the lab which is to serve as an instrument to enable learning by doing. This makes us want to propose a triage kind of model, especially in labs, where one of the TAs would help the students with their immediate questions, before they see the other possibly more experienced TA, to get more in-depth help, and the pair of TAs can learn from each other’s experiences.

We also had our participants suggest that knowing the student helped them chose an appropriate method to cater to the individual student’s learning needs. We could take this finding forward, and graduate our TAs, at least some of them, along with their students, following a cohort like community. This model could also use pair of TAs, where one of them is experienced with the students while the other is familiar with the students, and their mutual experiences can be used to benefit the learning of the students.
Bibliography


Appendices
MEMORANDUM

DATE: October 29, 2015
TO: Steve Harrison, Ayshwarya Saktheeswaran
FROM: Virginia Tech Institutional Review Board (FWA00000572, expires July 29, 2020)

PROTOCOL TITLE: Senior teaching assistants mentoring Junior Teaching assistants in an introductory programming course setting

IRB NUMBER: 15-952

Effective October 28, 2015, the Virginia Tech Institution Review Board (IRB) Chair, David M. Moore, approved the New Application request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

http://www.irb.vt.edu/pages/responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:

Approved As: Expedited, under 45 CFR 46.110 category(ies) 6,7
Protocol Approval Date: October 28, 2015
Protocol Expiration Date: October 27, 2016
Continuing Review Due Date*: October 13, 2016

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:

Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals/work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.
If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.
VT IRB 15-952 Amendment Approval Letter #2

Office of Research Compliance
Institutional Review Board
North End Center, Suite 4120, Virginia Tech
300 Turner Street NW
Blacksburg, Virginia 24061
540/231-4606 Fax 540/231-0959
e-mail irb@vt.edu
website http://www.irb.vt.edu

MEMORANDUM
DATE: November 5, 2015
TO: Steve Harrison, Ayshwarya Saktheeswaran
FROM: Virginia Tech Institutional Review Board (FWA00000572, expires July 29, 2020)
PROTOCOL TITLE: Senior teaching assistants mentoring Junior Teaching assistants in an
introductory programming course setting
IRB NUMBER: 15-952

Effective November 4, 2015, the Virginia Tech Institutional Review Board (IRB) Chair, David M Moore,
approved the Amendment request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved
protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the
IRB as an amendment request and approved by the IRB prior to the implementation of any changes,
regardless of how minor, except where necessary to eliminate apparent immediate hazards to the
subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse
events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

http://www.irb.vt.edu/pages/responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:
Approved As: Expedited, under 45 CFR 46.110 category(ies) 6,7
Protocol Approval Date: October 28, 2015
Protocol Expiration Date: October 27, 2016
Continuing Review Due Date*: October 13, 2016

*Date a Continuing Review application is due to the IRB office if human subject activities covered
under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:
Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant
proposals/work statements to the IRB protocol(s) which cover the human research activities included
in the proposal / work statement before funds are released. Note that this requirement does not apply
to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and
which of the listed proposals, if any, have been compared to this IRB protocol, if required.
If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.

<table>
<thead>
<tr>
<th>Date*</th>
<th>OSP Number</th>
<th>Sponsor</th>
<th>Grant Comparison Conducted?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Date this proposal number was compared, assessed as not requiring comparison, or comparison information was revised.
VT IRB 15-952 Amendment Approval Letter #2

MEMORANDUM

DATE: November 10, 2015

TO: Steve Harrison, Ayshwarya Saktheeswaran

FROM: Virginia Tech Institutional Review Board (FWA00000572, expires July 29, 2020)

PROTOCOL TITLE: Senior teaching assistants mentoring Junior Teaching assistants in an introductory programming course setting

IRB NUMBER: 15-952

Effective November 10, 2015, the Virginia Tech Institution Review Board (IRB) Chair, David M Moore, approved the Amendment request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

http://www.irb.vt.edu/pages/responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:

Approved As: Expedited, under 45 CFR 46.110 category(ies) 6,7
Protocol Approval Date: October 28, 2015
Protocol Expiration Date: October 27, 2016
Continuing Review Due Date*: October 13, 2016

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:

Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals/work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.
If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.
VT IRB 15-952 Amendment Approval Letter #3

MEMORANDUM
DATE: March 30, 2016
TO: Steve Harrison, Ayshwarya Saktheeswaran
FROM: Virginia Tech Institutional Review Board (FWA00000572, expires January 29, 2021)

PROTOCOL TITLE: Senior teaching assistants mentoring Junior Teaching assistants in an introductory programming course setting

IRB NUMBER: 15-952

Effective March 30, 2016, the Virginia Tech Institution Review Board (IRB) Chair, David M Moore, approved the Amendment request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

http://www.irb.vt.edu/pages/responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:
Approved As: Expedited, under 45 CFR 46.110 category(ies) 6,7
Protocol Approval Date: October 28, 2015
Protocol Expiration Date: October 27, 2016
Continuing Review Due Date*: October 13, 2016

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:
Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals/work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.
<table>
<thead>
<tr>
<th>Date*</th>
<th>OSP Number</th>
<th>Sponsor</th>
<th>Grant Comparison Conducted?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Date this proposal number was compared, assessed as not requiring comparison, or comparison information was revised.

If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.
VT IRB 15-952 Amendment Approval Letter #4

MEMORANDUM

DATE: April 11, 2016

TO: Steve Harrison, Ayshwarya Saktheeswaran

FROM: Virginia Tech Institutional Review Board (FWA00000572, expires January 29, 2021)

PROTOCOL TITLE: Senior teaching assistants mentoring Junior Teaching assistants in an introductory programming course setting

IRB NUMBER: 15-952

Effective April 11, 2016, the Virginia Tech Institution Review Board (IRB) Chair, David M Moore, approved the Amendment request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at: http://www.irb.vt.edu/pages/responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:

Approved As: Expedited, under 45 CFR 46.110 category(ies) 6,7
Protocol Approval Date: October 28, 2015
Protocol Expiration Date: October 27, 2016
Continuing Review Due Date*: October 13, 2016

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:

Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals/work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.
**Date**  | **OSP Number** | **Sponsor** | **Grant Comparison Conducted?**
--- | --- | --- | ---
--- | --- | --- | ---
--- | --- | --- | ---
--- | --- | --- | ---
--- | --- | --- | ---
--- | --- | --- | ---
--- | --- | --- | ---

*Date this proposal number was compared, assessed as not requiring comparison, or comparison information was revised.*

If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.
Data from Observations

<table>
<thead>
<tr>
<th>Question</th>
<th>Interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>62% problem coverage</td>
<td>TA read out the instruction specification “bullets at a constant speed of 10”</td>
</tr>
<tr>
<td>How would the TA have done it?</td>
<td>TA suggests a logical alternative (for greens).</td>
</tr>
<tr>
<td>Web-CAT problem coverage</td>
<td>TA reads the code and spots an if condition with multiple if statements</td>
</tr>
<tr>
<td>exception</td>
<td>TA goes through the exception thrown and looks at the problematic line.</td>
</tr>
<tr>
<td>setSpeed method</td>
<td>TA explains getters and setters generally and ties it to the setSpeed method.</td>
</tr>
<tr>
<td><em>0% problem coverage</em></td>
<td>TA goes over web-cat feedback</td>
</tr>
<tr>
<td><em>0% problem coverage</em></td>
<td>TA emphasizes the constructor requirement.</td>
</tr>
<tr>
<td>up and down dPad methods don't work</td>
<td>TA asks to show the code that makes asteroid move. (Where do you move them?)</td>
</tr>
<tr>
<td><em>asteroids moving too fast</em></td>
<td>TA explains the difference between (simply) calling a method and using the return values that the method returns.</td>
</tr>
<tr>
<td>I am getting a null pointer exception</td>
<td>Student executes the code and demonstrates the exception</td>
</tr>
</tbody>
</table>

Student: “where do I put it?”

TA: “where did you put it in the asteroid class?”

Student: “oh I get it”.

**How would the TA have done it?**

S first explains all his code.

TA suggests a logical alternative (for greens).

S executes his program.

TA checks at code again, TA: “always comment” and stresses on the importance of good documentation practices.

Web-CAT problem coverage.

**“sometimes bullet is removed when it hits the asteroid”**

TA explains what the web-CAT error means.

TA asks a question to help student locate the problem “where are you setting it?”

TA asks the student some more questions such as “what happens at 0,0?”

**“0% problem coverage”**

TA explains what the web-CAT error means.

TA asks a question to help student locate the problem “where are you setting it?”

TA asks the student some more questions such as “what happens at 0,0?”

**“0% problem coverage”**

TA goes over web-cat feedback.

TA asks about the “this” keyword.

TA explains in general about what a null pointer is.

TA then explains that “this” wasn’t initialized.

TA: “can we take a look at your tests?”

TA then asks the student what happens when the method is called, and what happens when the ship is not added to the world.

TA also encourages to observe by playing the game.

**“asteroids moving too fast”**

TA asks student to show the code that makes asteroid move. (Where do you move them?)

TA explains the difference between (simply) calling a method and using the return values that the method returns.

TA also questions the student understood why it was happening. (“did you understand why it was happening?”) Student: “yeah I do”.

**“0% problem coverage”**

TA goes over web-cat feedback.

TA asks about the “this” keyword.

TA explains in general about what a null pointer is.

TA then explains that “this” wasn’t initialized.

TA: “can we take a look at your tests?”

TA then asks the student what happens when the method is called, and what happens when the ship is not added to the world.

TA also encourages to observe by playing the game.

**“0% problem coverage”**

TA goes over web-cat feedback.

TA asks about the “this” keyword.

TA explains in general about what a null pointer is.

TA then explains that “this” wasn’t initialized.

TA: “can we take a look at your tests?”

TA then asks the student what happens when the method is called, and what happens when the ship is not added to the world.

TA also encourages to observe by playing the game.

**“0% problem coverage”**

TA explains all his code.

TA suggests a logical alternative (for greens).

S executes his program.

TA checks at code again, TA: “always comment” and stresses on the importance of good documentation practices.

Web-CAT problem coverage.

**“sometimes bullet is removed when it hits the asteroid”**

TA explains what the web-CAT error means.

TA asks a question to help student locate the problem “where are you setting it?”

TA asks the student some more questions such as “what happens at 0,0?”

**“0% problem coverage”**

TA goes over web-cat feedback.

TA asks about the “this” keyword.

TA explains in general about what a null pointer is.

TA then explains that “this” wasn’t initialized.

TA: “can we take a look at your tests?”

TA then asks the student what happens when the method is called, and what happens when the ship is not added to the world.

TA also encourages to observe by playing the game.

**“0% problem coverage”**

TA goes over web-cat feedback.

TA asks about the “this” keyword.

TA explains in general about what a null pointer is.

TA then explains that “this” wasn’t initialized.

TA: “can we take a look at your tests?”

TA then asks the student what happens when the method is called, and what happens when the ship is not added to the world.

TA also encourages to observe by playing the game.

**“0% problem coverage”**

TA goes over web-cat feedback.

TA asks about the “this” keyword.

TA explains in general about what a null pointer is.

TA then explains that “this” wasn’t initialized.

TA: “can we take a look at your tests?”

TA then asks the student what happens when the method is called, and what happens when the ship is not added to the world.

TA also encourages to observe by playing the game.

**“0% problem coverage”**

TA goes over web-cat feedback.

TA asks about the “this” keyword.

TA explains in general about what a null pointer is.

TA then explains that “this” wasn’t initialized.

TA: “can we take a look at your tests?”

TA then asks the student what happens when the method is called, and what happens when the ship is not added to the world.

TA also encourages to observe by playing the game.

**“0% problem coverage”**

TA goes over web-cat feedback.

TA asks about the “this” keyword.

TA explains in general about what a null pointer is.

TA then explains that “this” wasn’t initialized.

TA: “can we take a look at your tests?”

TA then asks the student what happens when the method is called, and what happens when the ship is not added to the world.

TA also encourages to observe by playing the game.

**“0% problem coverage”**

TA goes over web-cat feedback.

TA asks about the “this” keyword.

TA explains in general about what a null pointer is.

TA then explains that “this” wasn’t initialized.

TA: “can we take a look at your tests?”

TA then asks the student what happens when the method is called, and what happens when the ship is not added to the world.

TA also encourages to observe by playing the game.

**“0% problem coverage”**

TA goes over web-cat feedback.

TA asks about the “this” keyword.

TA explains in general about what a null pointer is.

TA then explains that “this” wasn’t initialized.

TA: “can we take a look at your tests?”

TA then asks the student what happens when the method is called, and what happens when the ship is not added to the world.

TA also encourages to observe by playing the game.

**“0% problem coverage”**

TA goes over web-cat feedback.

TA asks about the “this” keyword.

TA explains in general about what a null pointer is.

TA then explains that “this” wasn’t initialized.

TA: “can we take a look at your tests?”

TA then asks the student what happens when the method is called, and what happens when the ship is not added to the world.

TA also encourages to observe by playing the game.

**“0% problem coverage”**

TA goes over web-cat feedback.

TA asks about the “this” keyword.

TA explains in general about what a null pointer is.

TA then explains that “this” wasn’t initialized.

TA: “can we take a look at your tests?”

TA then asks the student what happens when the method is called, and what happens when the ship is not added to the world.

TA also encourages to observe by playing the game.

**“0% problem coverage”**

TA goes over web-cat feedback.

TA asks about the “this” keyword.

TA explains in general about what a null pointer is.

TA then explains that “this” wasn’t initialized.

TA: “can we take a look at your tests?”

TA then asks the student what happens when the method is called, and what happens when the ship is not added to the world.

TA also encourages to observe by playing the game.

**“0% problem coverage”**

TA goes over web-cat feedback.

TA asks about the “this” keyword.

TA explains in general about what a null pointer is.

TA then explains that “this” wasn’t initialized.

TA: “can we take a look at your tests?”

TA then asks the student what happens when the method is called, and what happens when the ship is not added to the world.

TA also encourages to observe by playing the game.
<table>
<thead>
<tr>
<th>How do I get the time?</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA: “what is a substring?”</td>
</tr>
<tr>
<td>Student doesn’t know.</td>
</tr>
<tr>
<td>TA: “how can you get digits from a date string, for ex: 10 pm?”</td>
</tr>
<tr>
<td>Student: “am confused”</td>
</tr>
<tr>
<td>TA: “think about the last characters”</td>
</tr>
<tr>
<td>TA explains how the string has “am” or “pm”as the last 2 characters, so the first n-2 characters would be the digits in the string.</td>
</tr>
<tr>
<td>So the substring method can be used to get the time alone.</td>
</tr>
<tr>
<td>S is then confused about how to set the time.</td>
</tr>
<tr>
<td>TA: “what does a set method do?”</td>
</tr>
<tr>
<td>Student: “to make new objects”</td>
</tr>
<tr>
<td>TA then tries to explain what happens when a new object is created and then asks “what the “this” keyword stands for?”</td>
</tr>
<tr>
<td>S: “what does the object do?”</td>
</tr>
<tr>
<td>TA then explains what happens when a method is invoked by an object in relation to “this” keyword.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comparing string to integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA: “what are you trying to do here (points at region of code)”</td>
</tr>
<tr>
<td>S: “returns”</td>
</tr>
<tr>
<td>TA: “what is the scope of this variable?”</td>
</tr>
<tr>
<td>TA explains the return statement, and how it would work inside an if-else construct.</td>
</tr>
<tr>
<td>S: “it has created an “appointment object,” with null description inside double nested for loops.”</td>
</tr>
<tr>
<td>TA: “think of it as a slot in your wallet.” TA also refers to previous lab for explaining object assignment.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>I am not sure what to do</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA: “what is the method supposed to do?”, “In real life how would you add an appointment?”</td>
</tr>
<tr>
<td>S: “I would add the appointment in a particular day and time”</td>
</tr>
<tr>
<td>TA: “as you need the time and a day and what the appointment is? how would you “get the time in the appointment (object)?””</td>
</tr>
<tr>
<td>TA: “hints about the getter method”</td>
</tr>
<tr>
<td>S: “the getTime method in appointment class”.</td>
</tr>
<tr>
<td>TA: “think of it as a slot in your wallet.” TA also refers to previous lab for explaining object assignment.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>addAppointment method</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA: “what is the functionality expected from addAppointment method?”</td>
</tr>
<tr>
<td>TA: addAppointment is supposed to add the appointment to the array and the world, in this case the weekly calendar.</td>
</tr>
<tr>
<td>S: “so should I add it to the world?”</td>
</tr>
<tr>
<td>TA: “yes”</td>
</tr>
<tr>
<td>S: “also, how would I assert that an object is at a said location?”</td>
</tr>
<tr>
<td>TA: “you can use the objects X and position”</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>getAppointment method</th>
</tr>
</thead>
<tbody>
<tr>
<td>S: “what exactly should I do in this method?”</td>
</tr>
<tr>
<td>TA: reads the requirement out of the lab specification and explains it.</td>
</tr>
<tr>
<td>TA explains to the student about getting an appointment from a calendar, in real world, and the breaks it down for implementation.</td>
</tr>
<tr>
<td>TA suggests that the student check for safe ranges for the calendar and then use it to index the array</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>addAppointment method</th>
</tr>
</thead>
<tbody>
<tr>
<td>S: reads the requirement</td>
</tr>
<tr>
<td>TA: “what are you trying to do?”</td>
</tr>
<tr>
<td>S: “I am trying to add an appointment to the array”</td>
</tr>
<tr>
<td>TA: “show me what the appointment is and where the array is?”</td>
</tr>
<tr>
<td>S: explains 1 part of the appointment as day and the second part as hour.</td>
</tr>
<tr>
<td>TA: “you need to know what day and what hour to put it on?”</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A question about overriding</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA: “if you already have a method, you can write a method with the same name that does something else and when you write the javadocs you have to use the @override tag”.</td>
</tr>
<tr>
<td>S: is asked to show the javadocs written for the method.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>How to remove the existing appointment</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA: “tell me your ideas”</td>
</tr>
<tr>
<td>TA prompts saying “what is the 2D array? where and what is represented as?”</td>
</tr>
<tr>
<td>S: “of the world is the 2D array, so I have to remove it from the world”</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>I am trying to understand inheritance</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA: uses a vehicle-&gt;car example. “are all cars vehicles?”</td>
</tr>
<tr>
<td>S: “yes”</td>
</tr>
<tr>
<td>TA: “are all vehicles bikes?”</td>
</tr>
<tr>
<td>S: “yes”</td>
</tr>
<tr>
<td>TA: “are all bikes vehicles?”</td>
</tr>
<tr>
<td>S: “yes”</td>
</tr>
</tbody>
</table>
"how would I write it"

TA: "ok, it returns an arraylist"
TA attempts to break it down.
TA: "what do you need to do with the given list ex { a, b, c, d }, you have to return a new list
[ d, c, b, a ]"
TA points out that size() needs to be used instead of length for arraylist.
S: "how would I reverse it ?"
TA: "get one particular string, do you know what to do with it ?"
S thinks out loud.
TA: "you would have to add this string to the new list"
S: "would that be newList.add(i) ?"
TA: "but i is an integer"
TA: "what does toReverse.get(i) give you ... in this case it would give you a string."
S: "i am not sure what the get method does." (he doesn't seem to get where and what to add)
TA: "the get method returns a string, the return value needs to be used somewhere ? this line does not do anything"
S: "???"
TA: "think about how would you get them in reverse?"
S walks through his code and restates what each line does.
TA: "can you not your existing for loop to traverse from end to beginning instead of
beginning to end "
S had used size(), and seems to be confused about starting at size()-1
TA explains for loop initialization.
S does not understand the index and how that relates to size() -1
TA: "if indexing starts at 0, the last index will always be ... ? "
S: "size() - 1"

"maps"

TA: "maps in programming are just key value pairs, "
you give it a key and it will give you a value corresponding to the key.
"You can think of it as a word in a dictionary, where word is the key and the meaning is its
associated value"
S: "so in this question, the map will be < string, person > ?"
TA explains how the key and value can be any data type.
TA: "look at the parameter, it is an arraylist of persons"
S writes down the method signature.
TA: "have you seen an interface in java ? ( lists -> arraylist , map -> hashmap )"
S: "you have to add the mappings. humans.get(i) is a person object"
S tries to "add" to the map, TA points out that maps don't have an add method and instead
they have a "put" method.
S: "so what will be the string ?"
TA: "the assignment is telling you to map the person's name to the person object "
S: "so the get(name) will be the string ? do i use humans.get(name), ?"
TA: "you cannot call get(name) on a list of persons, u have to call it on a single person" -
types Person with an uppercase P , TA points out that it cannot be called on a class.
TA asks the student to think about how to get a single person and that the student has
already done that.
S: "as if humans.get() , "
TA: "yes, so do get(name) on it "
S: "getName() instead of ?"
TA: "no, getName() on top of the person that is returned by humans.get() "
types the key and the value
TA asks if the student understands why that would work.
S does not seem to get it
TA breaks it down again. "Person person = human.get(i); String
nameOfPerson=person.getName(); "
S: "the lowercase p is the object "
TA recommends using the for-each loop ,
S: "can you explain it ?"
TA: "you wouldn't need a separate counting variable (i) . "
TA explains the left side is n object and the right is a collection of such objects
S: "so what would be the equivalent of person p in (code) ? "
TA: "it's the same as accessing the list with the index "

"how do I create forward command object"

TA: "for hashmaps, we have to put in 2 objects .. right ... how would i create a forward
command object?"
S: "how would you create any object?"
TA: "using the new keyword. ?"
S: "that's correct!"
S: "how do i put it in a rocket ?"
TA: "which class are you writing this code in ?"
S: "the rocket class"
TA: "how would you refer to a rocket object within the rocket class?"
S: "should be using the "this" keyword?!
TA: "yes!

"which constructor should I call?"

TA points at the code and starts explaining.
S: "how would you know which constructor to call ?"
TA explains the requirement and then points at the code for corresponding constructor and
why that should be used.

"How to add rocket to forward command"

TA starts by asking questions -> "what class are you in ?"
S: "rocket"
TA: "you want to add the rocket that you are currently writing to the forward command"
S: "speed this rocket"
TA: "why this rocket?"
TA: "what does the "this" reference mean ?"
S: "is not giving an answer"
TA: "this is the reference to the calling rocket object, and that is what you should be using to add this rocket to the forward command"
"I am trying to run the test case and am getting null pointer exception on return, I feel like I set it up wrong."

TA: "Let us take a look at your code. Scroll down."

"Can you explain to me why I am getting a null pointer exception?"

TA: "All my test cases give me a null pointer exception."

TA looks at the code, "scroll down please." TA finds the same issue as previous student. TA explains how the default and the parameterized constructors work.

"How do I implement the command?"

TA looks at the students code. TA: "So, you are not supposed to be calling the execute method on all instances, instead call the execute method on whatever object is returned?"

S: "I don’t get it.

TA: "Okay, let's do it step by step."

TA asks the student about the map concept. TA: "So each key has a value of type object, call execute on that object; you don't even need the if statements now because you are going to find out which object has that command and call execute on it."

TA suggests looking at the map API. TA: "You should call the get method using the key, it takes a key and what would that be in this case?"

S says he is still unclear.

TA then explains maps as a concept, and explains how to use execute on that.

S: "Do I have to do this once for every single command?"

TA: "All forward, left, and right are all commands, so you can simply call a command object."

S: "I think it is "Command cmd, new Command()" and TA: "You don't need to create a new object, you can simply use the return object." 

"I am getting a stack overflow error."

TA looks at the code, "Scroll down please."

TA finds the same issue as previous student. TA explains how the default and the parameterized constructors work.

"Every time I add a rocket to the world, it disappears!"

TA: "Let us look at your rocket class."

TA: "As long as your test cases pass, you are good, it might be just a weird greenfoot issue."

"I am getting a null pointer exception."

TA looks at the code, "I suggest breaking up this part of the code, save input.next() to a variable."

S tries that and the issue still exists.

TA: "First of all you don’t need these command objects."

TA goes on.

TA: "Why are you creating a new object? Should you be sending this rocket?"

"I am getting a null pointer exception."

TA looks at the students code. TA traces the line number back to forward command class at line number 127.

TA: "You never initialized it, you only declared it."

TA: "Also rename your parameter name to be different from your field name."

TA asks TA to repeat, TA: "You need to initialize the rocket object."

"Null pointer exception."

TA looks at the code.

TA tells the student that he has redeclare it and initialized a different object. student fixes it.

TA: "Now your null pointer is because you never initialized this (points at code) object."

TA creates a new rocket, TA: "Why would you need to create this rocket? I just assigned the returned rocket to your rocket."

"Null pointer exception."

TA looks at the code.

TA asks the student about the code being thrown from the execute method and asks to see that method in all the classes (forward, left, and right).

TA: "Let us go back and look at rocket class."

TA: "Can you put rocket and rocket test class next to each other?"

S does as asked.

TA: "Compare the two, look at the forward command and what you are adding to your map?"

"webCAT is not happy, it says actor doesn’t move as expected."

S: "Maybe it moved out of the world?"

TA: "To go to forward command."

TA: "Open the left command class, is the rotation -90 or 90?"

TA: "Open rocket class."

TA reads the code out loud.

TA: "I make if the if statement that is added to check for null."

S clicks on act 18 times, to see what happens.

TA asks to see rocket class again.

TA reads out loud again.

S asks if he should add 2 if-else statements, instead of his current code.

TA explains: "That wouldn’t work."

TA: "What is it supposed to do if it gets a null? I assume I am thinking..."

S: "Ohh! I think we are supposed to add more commands."

TA: "And I have to use capital."

TA: "If the commands are sometimes beginning with uppercase letters."

S: "Okay, will I need more test cases?"

TA suggests looking at the interface and the parameters.

"Null pointer exception."

TA looks at the code, and states: "The default constructor has all the code, and the parameterized constructor is called by the tests."

"webCAT wants me to provide javadocs. I am not sure where."

TA: "Can you show me what web-CAT says?"

TA: "Okay, so you have to write javadocs for both the interface and also the methods."

"It says, I can’t use this move method."

TA points out that the move method takes a parameter by asking: "How much do you want it to move?"
"actor does not move"

S had used rocket.move(-90).
TA asks to see the rocket class.
Rocket constructor in the students code using a parameter.
TA points to the lab assignment and says "maps should be initialized and not s a parameter".
S makes the changes.
Now a null pointer exception is thrown.
TA: "why did you put all this code in the default constructor?"
S: "i don't know, is it nowhere it should be?"
TA: think about which constructor is going to be regardless of if there is a parameter or not?
TA goes over how parameterized constructor will be called even when you dont supply an argument.

"stack overflow error"

"I am getting an exception"

"Null pointer"

S had used map.get(input.hasNext()).
TA asks to see te rocket class.
Rocket constructor in the students code using a parameter.
TA points to the lab assignment and says "maps should be initialized and not s a parameter".
S makes the changes.
Now a null pointer exception is thrown.
TA: "why did you put all this code in the default constructor?"
S: "i don't know, is it nowhere it should be?"
TA: think about which constructor is going to be regardless of if there is a parameter or not?
TA goes over how parameterized constructor will be called even when you dont supply an argument.

"I am getting a null pointer exception and I am not sure why"

"what does this do?"

"I am getting a null pointer exception"

"the compilation is very slow"

"I got null pointer exception, while trying to run these tests"

"I cannot get this to face this direction"

"I am trying to get my test methods to run, but I am getting a null pointer exception"

"I am trying to get my test methods to run"
"I have a question about this method"

TA asks to open the map API and asks to look at the put and get methods specifically. TA then explains how these methods work. As the student tries to use the put method TA also points out that the code is in the wrong constructor and the map instance is local to the the constructor because of its redeclaration.
S: "where is the hashmap coming from ?"
TA uses the list arraylist example to explain how hashmap is an implementation of map interface.
TA then questions the student about where he would create the ForwardCommand objects.

"web-CAT is throwing an error"

TA reads the error and explains that the issue is with naming, student did not use camel case for method names. TA also sees that the student has created more objects than necessary for forward... left right commands and points that out.

"I am getting an error, what does input.next() do"?

TA explains that the next command is fetched through input.next(). Student has map initialization code in the default constructor of the rocket class. TA points out that the tests call the parameterized constructor and thats why the object is null.

"null pointer"?

TA asks to look at code, side-by-side, of the rocket class and the ForwardCommand class.
TA: "why is the rocket field inside the rocket class?"
S: "so should i put this field in there ?"
TA: "yes!" (one more NPE)
TA asks to look at the ForwardCommand class again, pointing at code.
TA tells the student that the rocket field needs to be assigned the parameter.

"I am not sure how to do this"?

Student has rocket object as a parameter inside rocket class’s constructor. TA tries to explain the this keyword, that calls the parameterized constructor, which takes a scanner parameter.
TA also notices the student has used if-else, on top of using maps, in the act method to check what object the map returns. TA: "remember all your ForwardCommand, LeftCommand and RightCommand are all command objects. S: "so can i just call execute on those objects?"
TA: "exactly"

"My program won’t compile"?

TA looks at the error message and points out that the Student is missing "implements Command" as part of the class signature in all the Command-class implementations.
S: "alright. did i make no changes... and the TA then explains what "implements" means.

"web-CAT says that this line is too long"

TA suggests breaking u the line and concatenating them using a + for the intersection method, do we need to clone the array?

"I am getting an error , what does input.next() do"?

TA explains that the next command is fetched through input.next(). Student has map initialization code in the default constructor of the rocket class. TA points out that the tests call the parameterized constructor and thats why the object is null.

"Null pointer"?

TA asks to look at code, side-by-side, of the rocket class and the ForwardCommand class.
TA: "why is the rocket field inside the rocket class?"
S: "so should i put this field in there ?"
TA: "yes!" (one more NPE)
TA asks to look at the ForwardCommand class again, pointing at code.
TA tells the student that the rocket field needs to be assigned the parameter.

"I am not sure how to do this"

Student has rocket object as a parameter inside rocket class’s constructor. TA tries to explain the this keyword, that calls the parameterized constructor, which takes a scanner parameter.
TA also notices the student has used if-else, on top of using maps, in the act method to check what object the map returns. TA: "remember all your ForwardCommand, LeftCommand and RightCommand are all command objects. S: "so can i just call execute on those objects?"
TA: "exactly"

"My program won’t compile"

TA looks at the error message and points out that the Student is missing "implements Command" as part of the class signature in all the Command-class implementations.
S: "alright. did i make no changes... and the TA then explains what "implements" means.

"web-CAT is throwing an error that this line is more than 80 characters long"?

TA: "how would you put 2 quoted strings together?"
S: "use the + ?"
TA suggests breaking up the original string into 2 smaller strings.

"I am getting a null in line number 67"

TA explains when a NPE would be thrown generally and questions where there might be a null.
S: "input might be null"
TA: "okay, so let us look at where input is declared and initialized."
S: "i had not assigned the parameter scanner to the input field.

"I am not sure what I did this right?"

TA looks into code.
S: "is there a better way to do it?"
TA is still going over the logic.
S: "i think it might work."
TA: "couple of things... 1. why do you require a count?"
S: "i don’t know. i couldn't think of how to compare these"
TA points to the example he wrote on the board.
TA: "if you encounter a condition where there is a mismatch in the very beginning itself, you can return a false right?"
S: "then makes the change."
TA: "you have to see how this affects the remaining conditions... and secondly, you are looking for a[0] in b and b[1] in a"?
TA asks for paper and pen, draws an example array to show the student how he is checking for different things.

"For the intersection method, do we need to clone the array?"

TA reads the instruction for the student.
TA explains the concept of creating a pointer, through the example of creating a shortcut on your desktop vs copying a folder.
TA then explains referencing the str array and why ‘contents’ needs to be referenced.
"web-CAT tests for intersection method, object intersects with itself case is failing" TA reads the test case description on web-CAT. TA: "show me all the tests that you have written" TA reads out the lab instruction for the intersection method. TA: "you have to consider cases when one array does not have the element at all" S: "would you say that you have to null the checked element in both arrays?" TA: "yeah, you think about it"

"can you explain to him why we use the contents method as opposed to the contents field?" student to GTA: "it works when I use the method, and not the field I don't know why" TA: "let me throw an example" S: "I am just asking you about the equals method, in the interest of time" TA: "if you are calling it on itself, you are getting null values on it itself, which is a problem" TA: "if you are modifying it in place..." S: "I got unprocessed beliefs, when comparing identical bags" TA: "have you checked for order differences" S: "I think for them" TA asks for paper and pen, TA: "there seems to be a flaw in your logic" (TA then starts explaining the logic on paper.) TA: "what is the condition for equality?" S: "can I change it to false?" TA: "forget about the code, let me talk about the equivalence" S: "I am not sure" TA then explains TA: "so two bags will be equal if they have the same elements the same number of times"

S: "in the intersection method, I am trying to loop through two different bags and get the contents." TA: "remember" S: proceeds to explain his code. TA: "the only thing you need to add is that you should also null it in both arrays" (TA then shows how that would work with an example)

"it is not letting me submit a test case" TA: "did you change anything in the code? because this usually happens when you mess with the project set up" TA then goes over the students code. (some issue with eclipse, a clean-up was required)

"I have the right idea down but I do not know how to put it in the new string" TA: "can you show me the test cases that are failing?" (Student opens the test cases) S: "I was wondering how to make bags over here" TA: "well, how would you make them over there?" S: "I was wondering how to make bags over here" TA then teaches the student how to use the debugger, explaining the stepping over function versus stepping into and how to "watch" a variable.

"I am trying to write a test case" TA points out the issue with the naming convention" S: makes the change TA: "I am not sure how to test the functioning of the intersection method since it returns a grocery bag" TA: "you could test the size of the returned intersection, if your equals method works, you could check the occurrence of a particular item" S: tries to ask with an example (adding an apple and another apple, then a banana) TA: "so it is not storing any state right, the first time you would add a minimum right, first 2 apples, 1 banana so min = 2 first, and then min = 1 banana" TA: "can you show me the test cases that are failing?" (Student opens the test cases) S: "I was wondering how to make bags over here" TA: "yes, let us do that, do you want me to help you with that?" S: "yeah" TA explains how to use the debugger on eclipse, he explains the stepping over function versus stepping into and how to "watch" a variable.

"I am trying to debug this. I have contents in my bag and it is not returning anything" TA: "It is probably a for loop thing, you are checking at index 10" S: "it is probably a for loop thing" TA: "is this the test that fails?" S: has the debugger open TA: explains the stepping over function TA: "do that again" TA: "you should put a breakpoint there" TA: "now we are in it, we can see what is going on, you can use the drop down to see the first item" TA: "so this case tests checks if the number of apples in one bag is the same as no of apples in a given bag" (S had written the case to return true without checking the entire list and the student figures this while using the debugger)

S: "I am trying to debug this. I have contents in my bag and it is not returning anything" TA: "it is probably a for loop thing, you are checking at index 10" TA: "you can tell me what does intersection length return?" S: "does it return the capacity?" TA: "you want to use size() on intersection. I see another problem, we can debug that if you want to" TA: "let us open the variables view" S: "and TA proceeds with debugging" TA: "are you sure you are running the tests on the correct grocerybag?" S: "yeah" TA: "step into the intersection method, keep going...so I changed to 1, which loc are we in now?" S: "I think we are going to the array again" TA: "what is your loop for?" S: "to look at the second array" TA: "what are you updating it?" S: "oh man, that is what I am missing!"
but I have the loop go through them using numbers"
TA: "what do you want it to do when it is true?"
S: "want it to continue looking"
TA: "why is your condition, checking for a condition that we do not care much about? it makes sense?"
S: "yes"
TA: "ok, so what are we actually trying to discover?"
S: "if contents are the same,"
TA: "so we care about when it is not matched, right?"
S: "as, I just check when it is not matched and return false there?"
TA: "yes"

S: "I am not sure what to do for the toString method"
TA: "Okay, so the Stack itself has a toString method"
TA: "what is your stack object? use ctrl+space to pull up options and see there is already a toString method, you can simply print it and check format it prints in, and see if that is what you need , you don't need to write one yourself"

S: "how do i test the equals method?"
TA looks at the student's code
S: "remember that you are not testing the code inside the equals method"
S: "but web-CAT is complaining that I have not tested these lines"
TA: "you should call your equals method to test it, so you should simply test whether this object is the same as that one"
S: "so can I simply say assert(af.equals(b))?"
TA: "yes"
TA then restates the purpose of test driven development and how one should write tests.

"I am getting a NullPointerException at line 144"
TA: "is that in your test case?"
S: "both in my .java file and my tests"
TA: "why have you set this to null?"
TA: "you first have to declare a variable , then you initialize it in the constructor , if not, you will get a null"

"Web-CAT says if I'm comparing an object to null, I got to use == and not equals"
TA: "set me double check, it looks good to me"
TA: "can you check the previous lab to see how we test the equals method for a null case, because we do that every lab"
S pulls up previous lab
S: "this is how we did it"
TA: "okay, so you want to create a null object and then compare it in the equals method and see if that solves your problem"
S: "I thought about it, but its the same thing pretty much"
TA: "it is just web-CAT, the code looks good"

toString method does not return as expected and I cannot figure why"
TA: "the toString method is supposed to return something , you have your stack flavors right chocolate, vanilla etc
You would want to print them separately, using the toString method"

TA explains the requirement and the goal of the assignment to the entire lab on the blackboard.
He states what the push pop and contains method are supposed to do.
TA then also explains his observations from grading the previous programming assignment.
TA: "your commenting needs to be more sensible, secondly, you have to avoid hard coding values into your code, and third, avoid single letter variable names"
TA: "for counter variables ?"
S: "that is fine, but not for other variables"
TA: "did you create them by right clicking and choosing [Unit tests]
S: "yeah"
TA: "observe the build and then the issue is fixed, TA also notes a possible NPE, and states it "you have declared icy1 and then you have redeclared it in set up, so the reference below is actually pointing to null"

"I am getting a null pointer exception in testEatScoop method"
TA: "generally you get a null when you try to call a method on a null object, so here, you have declared the object and haven't instantiated it, so it is null Whenever you get a null, you got to check all the objects in the line where the error is thrown"
S: "okay, also web-CAT says I have not tested it, I am not sure how to test it in the for loop"
TA: "just test it with a bunch of items, say 4 items, so that the loop gets executed"

"the scoop method is printing backwards like left to right"
TA: "what do you mean left to right, stacks are usually top and down?"
Student tries to explain that left is the top most (most recent) item that went into the stack, and then the javadoc confused, he had mislabeled the print method's documentation for the scoped methods.
TA: "couple of things first, your scoop method is supposed to simply return the top most item on the stack and your toString method is supposed to do the printing. Again, you don't need to implement a toString from scratch you have to use the available toString method of the stack , for your IceCream class.
TA then tries to make sure all the other javadocs are in the right place.

"the toString method is throwing an emptyStack exception but I have the loop go through them using numbers"
TA looks into the code
S: "I am using flavors.empty(), what is the difference between that and .isEmpty() method?"
TA: "really, you are not expected to implement the toString method from scratch"
But the student and the TA are still talking through the code.
TA: "your code will work only the first time, since you are popping the elements as you go through them one at a time, at the end your stack will be empty?
Student explains that he has made a copy of the stack and that is what he used to push/pop the elements.
TA: "but that is still pointing to the same stack, it is not a deep copy"
TA once again: "honestly, you are only expected to use the stack's toString to implement yourIceCream's toString"
The student did not get that the first time, and exclaimed that the requirement was extremely straightforward then !
"If we call search on a stack and it does not have that element, what does it return?"
GTA: "I will have to look it up"
UTA joins in: "you can simply use the contains method of the stack for your iceCream stack"

"How do you return something that has already been removed from the stack?"
TA: "pop method does that for you, it will return the element as it removes it"
GTA was saying: "you have to store it somewhere before removing it, otherwise it will be too late"

"I am trying to figure out what is wrong with my test case, it is expecting a 0"
GTA: "what is on the left (of the assert), is what it expects"
S: "ah"
TA and student go over the code.
TA: are you updating numScoops?
S: "no, I will put it in there"

"I am trying to go through all the strings, do I have the right implementation?"
TA: "are you doing this for the contains method?"
S: "yes"
TA: "you could simply call the stack class's contains method to implement your iceCram stack's contains method"
S: "I'm confused"
TA: "you are simply making a specific stack, it is still a stack"
TA: "lets see oh, you just needed to save the changes"

"for stacks, I am trying to go through all the strings, do I have the right implementation?"
TA: "what does your toString method look like?"
S: "I did not make one"
TA: "you have to make one, that calls the stacks toString method"

"how would I get the contents of the stack??"
TA: "unlike lists, you can't access all the items in the stack, you have to keep popping them from top to bottom"

"My test is giving me a null pointer exception"
TA: "did you initialize the object?"
S: "boils like this is not used"
TA: "It is because you missed setup, it has to be camel case, setUp, so nothing got initialized"
S: "I see"
TA tries to find out more.
UTA looks at the code too.
S makes the changes and the error persists. this tie he gets "initialization error for jEvents"
TA tries to find out more.

"if flavors is null, do this, else do nothing, how do I do that?"
TA looks at the code so far and asks "what happens now?" the S explains, and he seems to have all the required behavior.TA questions about why he asks.
Web-CAT has complained that not all possible case are tested.
TA: "oh are you testing null?"
S: "yes"
TA: "I do so do that, and that should fix the problem"
S: "how exactly will I test that?"
TA: "just check if the number means the same before and after, or the peek returns the same element after adding null, that would be a good way to do it"
"I want to know if the logic is right."

TA: "Can you go back to your code, show me the checker class?"
S: "Are you sure?"
TA: "Is there a reason why you are not sure?"
S: "I do not know if my logic is right?"
TA: "So what you can do is, start writing comments in the code, that will tell you what each step does."
S starts stating each part of the logic
TA: "Don't tell me, document it."
S starts typing documentation
TA: "Tell me what might be better named than ch and ch1?"
S starts changing the names, in multiple places. TA intrudes and recommends the refactor and rename option on eclipse.
TA then does an autoformat
S: "how did you do that?"
TA: "You forgot to do that."
S: "Is this the best place to check your base case?"
TA: "Provide an example and ask what would happen, in that case."
S: "But that is not a palindrome, is it?"
TA: "Can we go back to your definition, let us look at where you will check this condition, actually, you tell me.
S: "Up here?"
TA: "You could."
S and TA are looking at logic.
TA: "What are you doing here?"
S: "I am checking for split characters and empty space."
TA: "What actually are you returning?"
S: "I am just making a new string here."
TA: "So what I am getting at is, where does it go back?"
S: "What do you mean?"
TA: "Think of the stack, what happens?"
S: "I go back here."
TA: "Let me explain more."
S starts stating each part of the logic
TA: "Once you return, where does the control go?"
Student does not seem to know.
TA: "Let us type in a bunch of print statements to see what happens."
TA: "What could happen after this guy prints, is it the next line that executes?"

"I am not sure why this is not working"

TA looks at code.
TA: "What would the index be if it is at 0 in here?"
S: "Oh, it would be 0."
TA: "You probably won't need that string you are creating."
S: "So, can I return all of them together?"
TA: "If you are returning the string, you can be considering it will be a string, how do you want to do that?"
TA: "Let's take an example."
S: "I want to begin at the end."
TA: "How would you get that char?"
S: "Can I use one of the string methods?"
TA: "Sure."
S: "So is that the CharAt method?"
TA: "Yes."
S: "For the last character, that will be (str.length - 1)?"
TA: "And why is that?"
S explains that the indexing starts at 0, and the last would be at str.length - 1.
S: "I would have to add that char and everything except the first."
TA: "If you are returning the string, you can be considering it will be a string, how do you want to do that?"
TA: "Let's take an example."
S: "I want to begin at the end."
TA: "How would you get that char?"
S: "Can I use one of the string methods?"
TA: "Sure."
S: "So is that the CharAt method?"
TA: "Yes."
S: "For the last character, that will be (str.length - 1)?"
TA: "And why is that?"
S explains that the indexing starts at 0, and the last would be at str.length - 1.
S: "I would have to add that char and everything except the first."
TA: "If you are returning the string, you can be considering it will be a string, how do you want to do that?"
S: "Ah, I can't remember the name of that method.
TA: "What is part of a set?"
S: "Subset... oh a substring!"
TA: "Oh, substring, not just substring right? and what do you want to pass in as parameter?"
S: "Index 0 to length - 2?"
TA: "Think about it, is the end index inclusive or exclusive?"
S: "It is exclusive."
TA then explains how that would become recursive. S and TA talk about what would happen when length is 1.
S: "Would that give me an error then?"
S: "Works it out on paper."
S: "Does the substring method give you an error if I call str.substring(0,0)?"
TA: "I believe so, how would we deal with that?"
TA then suggests handling that as the base case.
TA: "What happens if I call this method with a null string?"
TA and student then take an example "hi" and S talks through her code.

"What is a static method?"

TA: "Can you go back to your code, show me the checker class?"
S: "Are you sure?"
TA: "Is there a reason why you are not sure?"
S: "I do not know if my logic is right?"
TA: "So what you can do is, start writing comments in the code, that will tell you what each step does."
S starts stating each part of the logic
TA: "Don't tell me, document it."
S starts typing documentation
TA: "Tell me what might be better named than ch and ch1?"
S starts changing the names, in multiple places. TA intrudes and recommends the refactor and rename option on eclipse.
TA then does an autoformat
S: "how did you do that?"
TA: "You forgot to do that."
S: "Is this the best place to check your base case?"
TA: "Provide an example and ask what would happen, in that case."
S: "But that is not a palindrome, is it?"
TA: "Can we go back to your definition, let us look at where you will check this condition, actually, you tell me."
S: "Up here?"
TA: "You could."
S and TA are looking at logic.
TA: "What are you doing here?"
S: "I am checking for split characters and empty space."
TA: "What actually are you returning?"
S: "I am just making a new string here."
TA: "So what I am getting at is, where does it go back?"
S: "What do you mean?"
TA: "Think of the stack, what happens?"
S: "I go back here."
TA: "Let me explain more."
S starts stating each part of the logic
TA: "Once you return, where does the control go?"
Student does not seem to know.
TA: "Let us type in a bunch of print statements to see what happens."
TA: "What could happen after this guy prints, is it the next line that executes?"

"What is a static method?"

TA: "Can you go back to your code, show me the checker class?"
S: "Are you sure?"
TA: "Is there a reason why you are not sure?"
S: "I do not know if my logic is right?"
TA: "So what you can do is, start writing comments in the code, that will tell you what each step does."
S starts stating each part of the logic
TA: "Don't tell me, document it."
S starts typing documentation
TA: "Tell me what might be better named than ch and ch1?"
S starts changing the names, in multiple places. TA intrudes and recommends the refactor and rename option on eclipse.
TA then does an autoformat
S: "how did you do that?"
TA: "You forgot to do that."
S: "Is this the best place to check your base case?"
TA: "Provide an example and ask what would happen, in that case."
S: "But that is not a palindrome, is it?"
TA: "Can we go back to your definition, let us look at where you will check this condition, actually, you tell me."
S: "Up here?"
TA: "You could."
S and TA are looking at logic.
TA: "What are you doing here?"
S: "I am checking for split characters and empty space."
TA: "What actually are you returning?"
S: "I am just making a new string here."
TA: "So what I am getting at is, where does it go back?"
S: "What do you mean?"
TA: "Think of the stack, what happens?"
S: "I go back here."
TA: "Let me explain more."
S starts stating each part of the logic
TA: "Once you return, where does the control go?"
Student does not seem to know.
TA: "Let us type in a bunch of print statements to see what happens."
TA: "What could happen after this guy prints, is it the next line that executes?"
Student explains about what he wants to check, he wants to check if a single character is a letter.

TA: "Is there a method in the API that checks whether it is valid?"
S: "Is it the isAlphabet() method?"
TA: "You are close, you will know when you see it."
S: "isLetter()"
TA: "Yes, so you are kind of cleaning up, before you do your recursion, you want a helper method. It might work but might be inefficient."
TA: "So your test isPalinrome, you get to clean up first and then call a recursive function. So it is the helper method that calls itself."
TA: "Also, you are not explicitly removing the char but building a new string."
TA: "Repit the helper method calling itself part one more time for the student to understand where the clean up happens and where recursion appens."

"How do I write this constructor?"

Web-CAT has thrown errors about the students code being public.
TA: "Questions student about what the 4 types of access specifiers."
TA: "What is private access?"
S: "No visibility beyond the class."
TA: "What is protected?"
S: "Subclasses can see protected fields."
GT: "Questions the UTA, when UTA joins the conversation.
GT: "Starts setting bunch of example classes with protected, private and public access and then questions, "Who can see what?"
S: "Tries to answer."
TA: "Then draws enclosed boxes, labels the innermost box private, and the next one package and the next protected and the outermost box public."
TA: "So what is encapsulation?"
S: "So I should have put it as private fields, thanks."

"How do I write this constructor?"

TA: "Typically, you would want to instantiate the 2 fields you have."
S: "I cannot use method in the constructor?"
TA: "What would you do then?"
S: "Does node.next exist, at the time of creation?"
TA: "I don’t know."
S: "You need to put in the data and give this node a non-null reference, but you don’t have a next node, although you can later make one."
S: "Why do I have a current node?"
TA: "With respect to your linked list, you have a node (that as a data and pointer to another node)."
S: "So the next node is the current node? No, that is the next node."
TA: "Yes."
S: "Takes it on a paper and then TA draws it on the board."
TA: "Your node is the current node, and holds a reference to the next node."
TA: "You don’t need reference to a current node, because that is the object with the data and the pointer to next."

Focus Group with TAs - Transcript

Researcher: So just feel free to answer these questions, with whatever examples you can think of based on your experience being a TA for CS 1114 and CS 2114 and the examples, if they are recent that’s really awesome but if you can think of anything that comes to mind when these questions are asked, please feel free to discuss at length and let us begin. What do you think are your responsible as TAs of an introductory programming course?

A1: To facilitate learning especially with introduction to possibly complicated topics. Researcher: and what exactly do you mean facilitate learning?

A1: we try to present things in different ways and try to do the teaching in different ways that the professor's may or may not do, for example the lab scenarios are what that we do a lot and often an and that is this very applied situation at least for 1114 so we have to help and teach that material in a different way then it were done with the professors, maybe more formally, maybe more with examples, but we literally have an assignment that we have to revolve our teaching on.so..

Researcher: okay so what about office hours?

S: (I) Think part of the thing is also a boat how do you help them get through their obstacles but in turn it's not directly providing them solution running through the steps that an experienced person would do in order to get to a solution. So it's more about showing them the entire process of how you go about doing something rather than showing them what i in confidences finally done.

R: I think building self-efficacy and confidence is really important in intro classes’ because the students haven’t had a chance with the difficult material to prove to themselves that they can do it so if you can encourage them like “yes of course. You are more than capable of solving this problem ”, then they will believe it themselves and be able to get past this intro classes.

Researcher: mmm..

G: I think that it’s kind of being implied that 1 of the responsibilities of TAs of introductory classes is also to make sure that no student I turned off of the subject or drop it or leave it because they felt like they were not able to handle it or they felt like teaching or their particular learning style was not catered to or they were feeling too lost.

Researcher: mmm..

A2: Adding to that, I think I remember a couple of instances from last semester that I told them like “do not judge computer science based on this course. This course is not typically what an introductory course should be like.

G: For 1114?
A2: "yeah, for 1114, because it's not at least I don't agree with the, that that's an introductory course to anything. (Small chuckles from the group.) But year that's what I've been telling people but a part of our responsibility comes into us into letting them helping them find resources on how to do it because there would be APIs that you know...

Researcher: mmm..

A2: I think we can read documentation and all that but to read documentation and try those APIs is pretty new to these students. Helping them to reach to the place where they can read documentation and write.

A1: That's an emphatic nod from my end. Help them help themselves.

R: Building off of that reminding them that you have to get through the basics in order to do the cool stuff, so they sometimes get frustrated that "this is so boring" or like "I don't want to actually do this for the rest of my life" and I'm like "yeah me neither, (group laughs), "that's why my research is not in basic Java" right. So I try and tell them about some of the things that I'm doing in my research or you know whatever else, maybe in my upper level classes so that they know if they get through the basics they can do more things that are more fun.

Researcher: right..

G: Yeah they be like "why can't we just use a library for this", and I will be like, "someone had to make a library, and if you don't know how to make the things you're using you shouldn't just use someone else's stuff that you don't understand"

Researcher: right.

A2: I've been doing it with jobs not research to involve them. Like I will say "how do you think eBay would do it" or "how do you think PayPal would do it?" or something like that, examples that I know of and I go with it.

S: Building on top of that, putting that across using real life scenarios on where they see that, for example, a linked list, people don't see that as a linked list, but a good.. like I try to put that in the this thing where they start seeing these things in day-to-day situations where, let's say you are downloading an app on your phone and the page that it gives, it can be thought of as a linked list that each time a new page comes in, it is a linked structure. So once they start seeing and appreciating the things they are doing everyday that's when they see more value in it.

A1: Actually, that Reminds me of one example that I used to help in communicating the importance of these kind of things like they just finished a program where they had a little kangaroo detect and go through a maze and I'm going to take to attend recent example of the Google cars and how they have to make very critical decision and actually very ethical decisions on what to do in different scenarios and I tell them "look you're doing programming these simple things, guess who's going to program and
make the decisions on these really hard ethical problems. It's going to be a team of programmers.

Researcher: Right. Right. Do you have anything else that you think are responsibilities of TA of an introductory course?

G(6:05): fair grading. I mean I hate grading but.. especially because recently I had a TA for one of my graduate-level courses I could tell that didn't read a single comment in my code even though we were told to put them all in there because if he had read the comment he would not have been emailing me like why doesn't it work he would have seen immediately that I have made a very simple input mistake because I looked at an old document instead of the latest updated document for the program. So I think it's really important to make the students know that you have been reading the code and reading their comments. Ah.. I like to respond like if they ask a question in their comments and it is rhetorical I like to respond to it anyway so that they know that I read it. I mean they feel like the work is actually being looked at and considered fairly.

Researcher: Okay...

A2: I have done that with funny comments like some of those kids right pretty funny comments. And you go back and reply to those comments.

Researcher: hmm.. So I guess the next thing that I wanted to talk about was when a student approaches you with a particular issue how do you determine what kind of approach you are going to take to help the student and what kind of factors go into determining it. Um okay this time I'm going to try doing this or I'm going to try doing something else.

S: First of all you try and gauge if the issue is at a conceptual level or an implementation level because there are some students who know the concept and they're just having a hard time implementing it but there are a whole lot others you don't even know the concept and are still trying to implement it. So if it's the latter, then you have to take a step back first and just try to run them through the ideas first before going forward. And I.. I do that a lot, second thing is you never give them the direct answer but you point them towards possible solutions and then let them think on their own and take it from there sort of Umm like hand holding till the point where they can this is sort of how I go about it.

R: And I think the way that you figured out if it's at a conceptual or at least for me, how you figure out if it's conceptual or implementation is, anytime they show me some code, I ask them first can you explain to me what you are doing so I can understand what they understand and then if I see that they actually know what they're talking about they just may be messed up somewhere then maybe just point them in that direction or if they are kind of not understanding any of it and then take a moment and review what we're actually trying to do and then trying to achieve that again.

S: mhm...
Researcher: Any other factors that sort of make you or help you determine what kind of approach to take with a particular student?

A1: Actually I was going to say something very similar, like that is it conceptual or nitty gritty.

G: And in applying the solution too, if they confused about what their own code is doing, I tried to teach him the debugger tool as soon as possible.
(S agrees)
.. Because it's just so much more convenient to what stuff happening do not have to debug the code myself, we'll go through it together and then discover the problem when it occurs hopefully but that helps more I think at an implementation level. For the conceptual level I like, I be like "do you learn more from pictures" is what I ask a lot and they say yes then I will take out an index card and draw pictures for them and I leave it for them and tell them "you could take this with you in case that helps you".

A2: Is this to 2114?

G: Linked structures, lots of those.

A2: But you have a debugger tool. I have a different experience with 1114. I tried helping kids to understand the error, error which is like a null pointer exception, and you tell them, like some of them can trace back to the lines where they have problems like following them, but most of them will come with the same null Pointer exception again and again and again To a point where you know I have asked them to go to stackoverflow search null pointer exception Here you go this is how you would do it I've reached that point. Remember there was one student last semester, there has been no one this semester, but last semester there was a student who came to me with the same null pointer exception every office hour. So in the end I kind of ended up giving him the answer to how to do it like Giving Him pointers to stack Overflow page but yeah ours is slightly different, no debuggers.

A1: Right... Right... Yeah, our debuggers I don't think work, (group chuckles) I haven't been able to figure it out. But I used to try to introduce if I can print statements....

Researcher: mhm...

S: Yup

A1: like "here, just print this out" "uhmm. Here are some places you can print it out " and actually we do uses a lot and I don't remember the term that was used but it's like hack like a hack-made or hacker programming Style where it's just ,you just try to fix the code so that it doesn't compile or that it does compile ,you just keep on hacking it even if you don't understand it so you just keep on adding "oh if I add this keyword in, will it stop yelling at me" ..
Researcher: mhmm...

A1: I mean obviously it doesn’t do what it’s supposed to do so we have to especially in 1114 we have to combat that. A lot. Just because they are willing to just do some little thing and hope that their solution all that the thing they want compiled compiled and it doesn't work. I tried to do that by doing code walkthrough...

Researcher: mhmm...

A1: Before they hit the compile button I say "okay do a change, let's write it down step by step what happens ,and just we're going to trace it together" and things like Oh, I try to tell them again and again, I think that just today I told one student like 3 times about three null pointers. Every procedure step.

Researcher: mhmm...

A1: Like ok "that says null pointer, what line is that on? Okay let's go to that line and then oh it's at this line means that there is an object that is not that doesn’t exist, so I mean it’s probably just repetition, but repetition helps but it’s not an easy thing to, it’s not an easy habit to kick off.

Researcher: uhm... What about things like maybe time of the semester or how much they have learned so far like does that change the way you would, like at the beginning of the semester there’s a different approach that you would take as opposed to at the end of the semester or.. ?

G: I think that just has to do with the content that we are teaching cause we don't start link structures until the middle of the semester so the beginning of the semester I don't really need to draw as many pictures because it's a lot easier to understand arrays conceptually and you don't necessarily, like I mean, you can draw out the array but most people understand the idea of sequential data without being forced to look at pictures.

R : I think one of the big factors that I use is actually knowing the student, they come to me I know that, maybe they generally do struggle with the concept or I know that they typically know the concept and probably just looking for like maybe a bug here and there especially since I had a lot of students last semester in 1114 so I can... If I had them in 1114 uhm...I can say... I can relate back to 11:40 like “do you remember when...

Researcher: uhm. Right right..

R : ..We learned about this “, you know, “how does that apply in this situation?” and I generally know them too, from last semester.. So that has helped.. A lot.

G: Followed some students...
R : Yes... That was the plan apparently..
Researcher: ahmm... Again does this change with being in lab or being in office hours does that change anything at all? Or does it still very much problem/student specific? Or have you noticed that kind of a difference?

S: Like what exactly do you mean? Changes as in?

Researcher: uhhmm. How you choose to approach on how you choose to help a student. does a change with if you are in lab or if you're in office hours?

G: I feel like in lab I'm a little bit more like I'll maybe give out the answer a little bit quicker, or like, "you definitely have a bug right here", because there is only 2 hours and...

(S: Yes) it's just that umm.. The labs can be so long that they don't have time to be perfect for like I won't care as much about variable names I wouldn't necessarily yell at them for that kinda thing, umm..if it's an office hours I definitely try to make sure that the concept is pretty solid and then ride them a lot more instead of being like "well obviously, you have problems with exceptions occurring"

Researcher: mhm.. Okay...

R: Yeah I would say that there's also a lot of ,like, it just depends on external pressure factors like if there are 15 students with their hand up I'm not most likely to sit down next to you and talk through with you, I'm more likely to kind of just be like "oh, a jump to that line, there's a problem here, fix that on your own " and then I gotta move on to someone else and probably come back and be like" did that work or did it not, are you still struggling with it? ",And then","let's sit down and discuss it together"

Researcher: okay...

A1: For us it's a little bit different because for like 1114 labs like to give the labs early on but that scenario you discussed was exactly me on the day before program was due, I got so many kids and I'm like "okay here's what.. Here's hear some ways that you need, go off" "okay next person, here's some things, do 'em", you know," fiddle with that", kind of thing. It depends on like how many people you have to help, I think...

G: Oh yeah 2 labs ago there was an issue, where ummm, I think the labs tend to be sometimes buggy themselves. We had an issue where the reference tests that that would be graded against what checking that the exception had a message input into it and it dint really protect the try statement so , or the tests so that , if they didn't put that message in their try statement, they would get a null pointer exception from the tests. And it's really difficult to explain that to them without being like "the test was looking for the message" so it was a lab so you know what, i'm just going to tell everyone- "just put it there", "it says right here in the lab, put a message in there and you'll be fine" whereas if it was in their projects I would be a little bit more a little bit more guiding about that.
Researcher: mmmm...mhm... Okay... uhm... So when do you know that you've helped a student enough or how do you know when the student has, like you know, enough to like figure it out? ?

S: when he leaves and never comes back ..
(group laughs)

Researcher: when he leaves and never comes back?

S: like yeah, doesn't come back at least for a while, on that particular thing, now that can have two implications there, I understand
That but then i'm a very positive person so I take it that way (group chuckles)..

R: Personality..

Researcher: mmm, when do you more like when do you stop to see if the student has just gotten the concept or like how long do you keep explaining you know that sort of..

G : I would stop maybe every couple of sentences and be like "does that make sense" and pause and watch them process it. A lot of times though they will signal when they think they've started to understand..

Researcher: Okay okay.. What about cases where they've made a mistake and you sort of help them through it and then that kind of means changing a lot more things or this trickling down and maybe fixing other things so how do you like srt of ..do you still go over it or do you wait for them to how to.. How does that happen? like how do you deal with that?

A1 : I usually break it down piece by piece so say like " oh there's a field and you should not have it public, just letting you know", duh duh duh duh duh and then " oh there's a null pointer exception in here, some ways to treat that" and then that might point to something else, and I will be like " oh and by the way for these things.." and then usually by the end I would say "okay so you have 3 things to work on you can just pick one of these three things, the first thing the second thing the third thing and then you can tackle each one of them and test those individually, so if I can just like almost compartmentalize like the error into a single thing they can do or a single thing that they can test or whatever then those are the, that's how I try to break it down so it's not as overwhelming, so it doesn't be like a spiral of terrible things to fix...

Researcher: Okay...

G : There's lab situations where webcat will be catching something that didn't catch in their own tests and almost always in that case I encourage them to try more edge cases or like " did you test it in this condition and this condition".. Sometimes I do look at what test they failed and then you could either tell or it's pretty easy to infer what they might have missed and be like you need to go test this thing and then I
don't.. I'm done.. " you test that and if you have a problem trying to fix it, when you
find the error come back to me, but..

researcher: Okay..

G: And then there is also cases when they come to me with buggy code and it's not just
not testing and they know there's a problem, if they are a student who I don't feel
struggle a lot, which is the view i default to, there's only a few students that I feel
struggle a lot by this point but, i'll fix the one major thing with them and if I see
other bugs I might mention.. "that's a good start" and be like "that mike not be
completely bug free but go test it, go try it out", but if it's someone who's
struggling that's the only time I sit down and go like " okay let's walk through
everything line by line to completely fix your method".

researcher: Any other thoughts?

R : Can you ask the question again? I'm sorry..(group chuckles).

Researcher: So I was asking if, ..

S: Thank you.. (group chuckles)

Researcher: I know it is.. (chuckles)

Researcher: when do you know that you have helped a student enough, like when you are
with them and helping when do you know that you can a student know when you’re with
them helping when you know you want to stop it, and have them figure it out or when do
you know they've gotten enough to process or or is that even the point that you would
stop?

R : I mean I think in some cases, you can't know that, like they would want to sit
there and and they would want you to tell you, tell them the answer right so in some
cases you kind of just have to push them out of the nest and go like "now you, I think,
you have enough information and I want you to go try it" but always remind them that
they have the option to come back to you, sometimes you just have to say " go to it on
your own and come back to me if you still cant figure it out”
(S agrees) that's usually my experience.
S : Yeah

R : I usually don't feel like this person will definitely go out and know exactly what
they're doing but I say "I think that you know enough now and come back"

A2: I do something similar, but I try with larger chunks, like I give them a big
Overview at the beginning for them to try out and if they come back again then I go
like little finer on the work and if they come back to their time on the 4th time I
just give up and go like," okay here's how you do", you’re not giving them the answer
but telling them steps that they could do.. The first time I tried to make it more
generic so that they could think for themselves and then I grow Restless as well when the semester goes along..

Researcher: okay.. (to S) you were gonna say something..

S; yeah, one thing that I do is sometimes I ask them to explain it back on what I explained to them, so sort of getting a reverse feedback on what their understanding of it is so. So that kinda helps in knowing if whatever I just tried to transmit, did it go through or got stuck..

A1 : I also do silly little examples code too, like tweak whatever they give me. or write some code and be like "okay so what does this do" matches very well to what the problem that they gave.

R: Sometimes if I have enough time like if they’re not busy they’re just sitting next to me I will ask them to write it in front of me so that I can see where the logic is and then maybe point out some things like" why did you make this decision ? you know and then they can explain it back and then maybe tweaked their concept a lil bit,

A2 : I just realized I asked them 'is it clear" that's my question "is it clear now?" but I don’t go in

A1 : But this is like the issue they mentioned for self-efficacy like just today I was trying to guide this one student threw a problem she’s like," oh should I.. should I write this here?.. should I write this code right here?" and all that and I was like "just write it down.we will you go over that" and she was like "should I put it here, what if I try writing this here, what should I .. "and I said " Just do it" but what if I put it here? " just write it down, once there’s something concrete on screen, we can go over it"

Researcher : How often does writing on paper happen?

A1: I wish I had paper with me but I write it on on Word, I will draw on OneNote or something..

G: Index cards are fantastic. They’re like bite-sized and you can just give it to them.

R: I write on paper and they hardly ever do I would say (S and G agree -"yeah, definitely"). I write on paper all the time I draw a lot of pictures on t a lot of pages in a notebook and a lot of pseudo code you know, they hardly ever touch paper.

A2 : I have a notebook full of my own notes, I did a piece off and give it (S agrees-"give it to them", others agree too)

G: Yeah, I draw a lot of pictures and I go through the pictures step by step if I feel like If someone who’s really struggling with the concept i’ll actually like.. I don’t really write full pseudocode, but I’ll write like "number one this is the thing that has to
happen" and draw a map and then "this is the second thing that has to happen" and try to differentiate it a lot more completely from the first thing.

A2: In our case that helps a lot you know because people have problems with maze and moving around and grid calculations and all that. So you give them a square grid And ask them to calculate with their hands and all that, that.. that helps in programming some of the programming assignments. And they do write that much, so so it's been pretty good.

A1: We do need to push them though.

A2: I Explicitly ask them to come in the front, in lab, explicitly ask them to come in the front where there is a chalk and a board. I draw the grid for them and hand over the chalk to them to write, like, "do it, do it with your hand and then we will think about the program later on"

A1: And there was some other code where they had to Navigate a thingy, and I'm like "no you need pencil and paper, just go. you find some pencil and paper and you draw where you think the thingy is supposed to end up and then see if it works" and then she had to do it again because she miscalculated but you know ...

Researcher: So I'm going to ask you all to give me an example of a time when you've been very clear as to "okay this is what this student warms and this is what I'm going to give and this is going to work" and it worked? Like "this is easy" like "I have always had this," sort of.

R: wait.. what ??

G common things ??

Researcher: uhhm.. where You specifically knew how to tell them, how to help them.. Like you didn't need to think about it. As soon as this came up you knew this is what you're going to do..

A2: I've had that when they try to initialize an object that they have not declared as a field and they get exceptions and see red everywhere and Its like you just haven't declared it anywhere and initialized it so that's a very common problem..

Researcher: So how exactly would you tell them that? Would you just tell him that you haven't declared it?

A2: Oh no.I would say "where have you.. no, where have you declared it"? Because that is a common issue that keeps popping up so you ask them where they've declared it and they will say " it's in the other class", the other file all together that they somehow copied the name from and then you try to explain you know the difference between classes and scope and then that kind of helps. So that's what I've been doing. It's
always been rhetorical questions to be asked, it's always "where is it" "how is it" or you know " how does this know what you're talking about" and...

Researcher: You're actually telling them the answer by asking them a question about it?

A2: Something related to it that could guide them.

A1 I think the most recent one that's because this one came up in 2 labs in a row back to back was using Java scanner, using the .next() method and its one of the scenarios where, they type n scanner.next() once and then scanner.next() again and again throughout the next checks. uhm.. it's usually when they have code written down and they go like "my code is not working, it's not doing all the things it is supposed to do, it's not scanning all of it, it's missing some stuff.. " and then from there it is like a code walk through step-by-step where I literally put a word, "input.next() is equal to this", and I go step-by-step. "okay line 12.. uhm...what is input.next()? what is that? okay, then what is the next thing?" so, that is automatic code walkthrough and then eventually they give me the "aha", like "ohhhhhhh"...

Researcher: hmm... Okay...do you guys have examples?

R: I would say.. SO I have TAd 1114 for a really long time and it seemed like, I loved it..I loved it so much, but it seemed like most questions that people would ask me, I would have at least seen it in the past 6 semesters at some point,

A2: That's quite a lot..

R : SO I could know instantly whereas now in 2114 I sometimes find myself like, "I have no idea how to answer this question.."I am trying to think of specific examples form 1114,because those were very automatic, especially project specific questions, I knew all the little nuances in them and

I knew exactly when they came to me with a problem like , this is what they are missing and the instructions, whereas in 2114, I am such a newbie that I have never seen the material before, I find myself having to ask other people like "have you seen this issue" or "how do you explain this to someone?" but I guess in 1114, the biggest example was when we were conceptually understanding for-each loops, I remember a lot of people struggling with that, and I remember myself struggling with that in 1114 and having to give them, the way that I solved it every time, was writing on paper, with actual like blanks, with semi colons in between and giving them a phrase to remember, for each something in some collection and writing out those words in English and then giving them the piece of paper so that they could fill in the blanks anytime they needed to write a for-each loop. that seemed to work every single time; uhm... but there are no tricks like that that I have for 2114, not yet..

Researcher: I'm sure you will have it the next time.

Aakaash: Did you have problems with the if and the while in 1114?
R : with?

A 2 : like if something or while something? instead of...

R: ..like the opposite?Oh yeah, kids..

A 2 : students keep on doing that.. I.. I ask them..I ask them.. while you have reached D2, you are hungry, like that means you take each step towards D2, like you know giving them practical examples, but then if you say if , if is only one time kind of thing , that had also helped a bit, but that , that's during the first two weeks of the class.If they come back later on in the semester with the same problem, then I get really really restless about it..

Researcher: you guys?

S : I think I tend to use that sort of a thing when people are not getting the syntax of something right a lot of times, where you try to connect it with an older syntax of something that they know..because of which they can use that as the model for building the current syntax right like lets say if they wanted ti create an iterator or just even define an array, things like that so,so how do you go about doing that, that's the thing that I see at a lot of times people struggle with at least at the beginning, or even every time a new concept is given to them, so like in the last lab we had this idea of a comparator, that was being introduced to them for the first time, so I saw a couple of people who could not even conceptualize how,how they would have a new comparator that can be used, and I just pointed them to the idea of it would be the same of how they were actually using an iterator on string sort of thing, So connecting it to a previous idea, is a little pattern, is one strategy that I use..

G : I know I do that with scoping too..it's a pretty common thing.. they go like "how come I cannot find the add method and why it was underlined in red?" and I will be like "hold your mouse over the error, it says method not found, it can't find it, why can't it find it ?" and I lead the student into like "where is the method?"they will tell me it is in a different class" then I will be like," go find the method, show me the tab to the different class, then I ll ask 'why can't the other method find it in the other class? and you lead them to like "I need to put an object name and then dot and the method name, for it to find it. "

Researcher: hmm.. Okay.. The counter example. the one where you have tried many many different ways and you probably did not get the concept across,It could have been for various reasons, maybe the student was just not prepared to take all that in, maybe he/she was way behind, or maybe it was just too tough to get across, or there was not enough time, I don't know., like for different reasons.. Can you think of an example when you tried various approaches and nothing worked.

G : would this be the one where we actually understand it though, cause there is always the case that you debug someone's code but you literally can't find the bug..
Researcher: hmm.

A1: or webCAT (S agrees). they will be like,"why is it failing on this test ?" and I'll be like, "I don't know, I am sorry." (G agrees)

Researcher (32:28): mmm.No more like when you knew what You wanted to get across but for some reason there was just this Gap and you could just not do it.

A2: Students unfamiliar with Boolean algebra it's difficult for them to cover all the test cases, that cover all the parts ,especially in 1114 some of the students will write If something or something or something and It will Keep on going so It will have like 2 ^ this many and.. and then to just explain you know that requires you just to have an understanding of how boolean systems work or At least you know some level of combinatorics.. so that has been quite difficult for me at least.. Not this semester because this semester they have changed the lab structure,they open It on Monday and the students have time till the lab time to submit it, but last semester lots of students had issues in getting full Score from web cat because they couldn't have full test coverage So that I don't think I was able to get across..

Researcher : but do you recall the type of things that you tried to tell them?

A2: I tried boolean, I tried combinations, like head and tail combinations,because it's also my favourite, like “if you had 2 coins, how many different combinations would you have? Now think about 3 coins.. Now think about“.. Some of them had 6 clauses in the if, like this this this this this this,so like how many would you require.so you know,try to help them understand full types of combinations,and it's not very easy to do, like We have been trained in boolean algebra since freshman year and it's been 5-6-7 years now,it takes time to learn, so in an introductory course, it's not something that .. and that’s why I did not quite try a lot , try 3-4 times, and then if they don't understand, I just tell them that if you have multiple combinations , you cannot test it this way..

Researcher: SO at that point, pretty much just tell them how to work with it ?

A2:Yes they wouldn't understand why they have to do it but We gotta tell them that they have to cover all of these cases...

Researcher: So that was one case where the concept itself was a little too much for them to understand right, because they have not dealt with 1's and 0's before,right?

A2: Yeah, because to construct a new knowledge you have to have a previous knowledge upon which you build it right.If you don’t have that previous knowledge, then you can’t quite build it in one go, and we hardly get what like 10 minutes with a student, you know, the way we are in a distributed lab,so that time is not enough to teach such concepts, over the week you may get those through but then not all weeks have same kind of challenges so that may not work.. So yeah..
Researcher: hmm. So any other time that you guys really tried and nothing worked, and nothing seemed to work?

R: I remember trying to describe mmm... like an interface, to students in 1114, and coming up with a bunch of analogies, with some contract,you know a lot of trying to come up with pictures or drawings or diagrams like this is an interface implement this interface you know, and ummm. For some students no matter what how I tried to explain it ,it just would not sink in..I.. you know after you come up with all those examples, it's like,I dunno how to explain it, because it is kind of an abstract concept,when they are struggling with things like you know if statements sometimes,I think the intro level courses go a little too fast for them and try and throw too much at them a once and they are struggling just to figure out how to write a while loop and you are trying to teach them what an interface is or an abstract class is,they don't even know how to read an API or find documentation, so..I think it can be tough if you.. Especially,you are in a course or a curriculum that does throw a lot at them, like the 114 class does..

A1: And that was a topic of discussion too when I taught it,we did throw. the curriculum did they throw a lot at them , and I learned why, the reason is to give them a lot of power, so that they get excited about it instead of drilling with the nitty-gritties, the details of learning nitty gritties, we give them a lot,and we let them do a lot,and then we later go in depth on them,I still have mixed feelings about that methodology, because of what you all cited that you don’t have enough time to go in depth and debug those questions. But I can atleast tell you that was the goal of that.. Of that particular course..

A2 : I would argue that it does exactly the opposite In fact I've had to tell students to not judge computer science or computer science Classes based on 1114 that's how badly it has gone I had a student who had come to me , said he can't make it in computer science, he is thinking of going to political science, that has happened to me last semester,and you know... throwing them a lot of.. And in 1114 it's not just one programming paradigm that we a are talking about, we are talking about helping them test cases,having those test cases cover a lot of parts, and in fact last semester.. This semester I haven seen much, but last semester it was to do test driven development, first write test cases and then go and write code, and that thankfully isn't here this time around, last semester it was a little much, you know when you have students struggling to write the basic code to make the jeroo run around, and you are asking them to first write the test cases and then see how you can write code, that becomes quite a lot of work.

A1 : That's unfortunately...

R : That's something I have never been able to explain to students, the would always ask why do I have to write the test first ,because we are asking you to.. because it's a reasonable way to write code..They would just ignored you because they don’t care, they don’t want to do it that way.So it's like no amount of you telling them they should do it that way is going to make them want to do it that way..
A2 : and and.. Amazon and you know industry examples, that kind of convinces them that it’s necessary but I don’t think they believe it as well, it’s not the cool stuff as well, like testing has never been the cool stuff for anybody, so..

G : they will come to me with buggy code or we will fix it together, and they will be like “does this work?” “or they will have just finished a hard method and they will be like “do you think, the will work?” I am like, “I don’t know, have you tested it yet?” they will be like, “No, I haven’t written my cases” “well.. GO test it !!”

Researcher : I don’t think that happens in 1114 right? Like people don’t ever come to you and say ...(A1 agrees) yeah ?

A1 : How labs are written are in the style of do the test first..

G : so does ours, Let me tell you it doesn’t sink in..

A1: (continues).it’s not... in part that’s because, the fact that developed the curriculum had a lot of good evidence that this is the best way to go, like everything, this is not the one paradigm that works, but there is evidence for it.that is why we do it.. If a student tells me “can i just write the code first?” I just tell them yes, because, if you feel.. If you want to write the method first, and then do the testing later, go ahead.. if that’s what makes sense to you.

G: I just make sure that they understand that when you are done with a class, you should test that class before moving on to classes that use that class, because then you get in this whole sticky chain.

Researcher: Yeah..

S: I think two things, one is just off theirs, which is, there really never know, or just tell it, communicating to them how much testing really is enough because for them the idea of testing is just something that has to be done with for them to get through and not really more about securing your code or seeing how correct your code is..

Researcher : So seeing the bigger picture?

S: instilling that idea is kinda tough...and the second thing where I have always noticed a big big robten is when they don’t understand the difference between what a reference is when you just need it to point it to a node or something and they end up creating a new node for that, so let’s say,, and they still do that, it’s like 3/4th through the semester and people still keep doing that and one of the reasons I see is because they don’t get into the internals of how things are organized in the memory. So for them, they don’t have this concept of what really a reference is. And no matter, even if I explain it to them, it seems fine for a while, but the concept kind of takes down after a while.

R : I find myself explaining things talking about pointers, and then I remember they don’t learn pointers until the next class that they will take, So I am like “yeah, you
have a pointer that is pointing to this. And then it's like I have to explain it to them like "In memory it's..." but then they don't know that yet.. (S agrees). They just know lines in the IDE.

G: Kinda use pointer and reference interchangeably because they don't know the difference right now, so glad I am not teaching them the next class. (R agrees)

Researcher: That kind of makes me want to go to the next question. How much is too much and how do you know it? Like for us pointers, we have lived with it for quite a while so it is sort of, it's in us... so we talk about it like It's just that but, to them it might not be the case. So much do you like do you think about so "I am probably saying a little too much. That I should be" you don't need to worry about this now" that sorta thing. Does that happen when you are helping them, does that come in the background?

A1: Kinda, the scenario you were mentioning, where you don't know what's exactly being covered, I try to be very brief, summary, because it's, I can't really be sure that the topic is that I, that revolves around a problem, was covered or not, or maybe the student didn't understand it or maybe they were sick that day, or maybe it just wasn't covered enough at all, so as the TA, I don't remember, what are they supposed to know? So I am going to give a very brief introduction of that and then also present this with "you might have already covered this, look through your notes" or "you will cover it next week or very very soon, I don't know when" but if it's a little too much, don't worry about it, you haven't covered it, it's okay. So it is pretty similar to the scenario you mentioned where you don't know exactly what was covered in class.

G: I think where I usually start to back off is like detailed explanation of the stack, and of managed memory, especially because they are not doing C yet, they will have to next year, but umm... I think they have covered the call stack in the classes some because, usually when I ask "do you know anything about the call stack" they can kind of say yes, but I don't like, sometimes I explain a little bit about scoping variables in terms of a... it's on the stack right now with the method. But I don't really get into that very much. Like "don't worry about it now it's just, it has to do with the way things are put in the call stack."

R: Yeah, there are some cases when I want to explain why something works but I, I. Like for instance like hmm.. they have an if statement, if you put the first thing and it's an AND then You know you can put something safely. After this one because it's what is the term you know, "in java you can do that but not in all languages" or things like java garbage collector, but that doesn't mean anything to them so I can just say like...

A2: (intervenes) Like the garbage collector for 2114? isn't that too early?

A1: We talk about that a little bit.

R: So it's kinda, it's like how do. "Do I have to set everything to null?" "No you don't because, the garbage collector will come do it for you.".. But then it's like
it's different in different languages (G adds), so what is Java, do I go into all of that language specific stuff, probably not..

G : Actually in the last lab, like 2-3 weeks ago, where they had this central node right, so they had this list that was a doubly linked list, where the central nodes where pointing out, so they asked me, will java garbage collector collect that because if you just dereference the head and the tail, every single one of those nodes, still has other things pointing to them. I think the answer is no right? I didn’t really know if to tell the student either, I am like “I am glad that you could actually get that because everyone else in this room is like looking at us like what.”

R: yeah...

G: Some Of them get the garbage collector,

R : and some of them have, sometimes, even to gauge of this might be bad, this might be like biased, but to gauge how much I should tell them, I ask,” are you a major or a minor “, or are you even going like far in this, cause maybe you don’t care and you are not taking anything really after this, I mean some people are their main focus is they are mechanical engineers and they are taking this to get a basic understanding of programming, but they don’t wanna know like the guts of all of it so but then if they say like “I am a major” then I am like “Okay, you will need this later” or “you need to know this” I don’t know. Maybe that’s biasing my stuent and preventing some from learning..

Researcher : based on with this discussion I was thinking like, do you think that it would help for maybe 11114 TAs to know what’s happening in 2114, like what are they put up for is...

R: I Know they moved me to 2114 Specifically so that I could give my input on what they learned in 1114 And say “they didn’t cover this” or “they didn’t cover this enough” or “most of the students didn’t get this when they did the first time, so let’s cover it again”. So I know that they, that’s why I got moved from 1114 to 2114. They are trying to have some, some sort of consistency across the program. I don’t know if backwards, or like I don’t know if knowing what they are going to learn is helpful or not.. Definitely knowing what they have learnt..

G : Yeah no... I didn’t go here for undergrad, everything I know about curriculum and other classes, is just stuff that I picked up off the street, I haven’t studied their course book or anything like, I know there is a 3114 but I really don’t know that there is 2504 that they take, where they do something with like systems stuff but I don’t really know what they are eventually going to learn.

A1 : Cal ribbens was monitoring one of my teaching sessions, and one of the things e did suggest was like in, while you are covering topics, to say,” oh there is more about this cool topic, it’s in 2114 or 3114”, so it’s more like a tad bit shadow to say hey coming attractions, these are cool things in this class, I also don’t remember which one is which. I just tend to guess,”oh Boolean algebra, we will definitely cover that
 sometime”, “you have to, you must, just don’t remember which one, but there is a lot of fun in this” but it’s usually little tidbits like little highlights, because you don’t want to overwhelm them with like the details. It’s more like it’s better to as them questions they do know, to build up their confidence, “okay so what’s a variable? What’s an object? Is this an object? Yes / no?” So it’s usually referring to what they know, rather than what they will know.

Researcher: hmm.. But yeah, I actually never thought about that, but what do they learn before or after..

G : I do feel like I did more curriculum based stuff, or asked are you a major or minor when I was in undergrad because I knew the course structure and I had taken all the courses and so I could ask are you a CS major or the other was like IM minor, or an IM major with a CS minor, especially in the video game design students, cause then I’ll be like “Okay look, this portion right here, learning is very relevant to game design engines” or be like “this portion is relevant to what you are going to learn in your IM courses in that program and don’t worry about the nitty gritty and the other confusing stuff” because after the C++ class, they would theoretically be done with the minor, just take the electives.

Researcher : So.. ahmm.. Can you remember a time when you’ve actually thought about helping the student by actually modelling the process of solving a problem like, for example, if it’s a Null pointer exception, then you are not only just trying to tell them what that is but how you would go about debugging it the next time they come across a null pointer exception. Do you think about that or how do you actually try and incorporate “so next time you have it, this is what you are going to do, so follow what I am doing right now”, sort of approach?

G : That is exactly what I am doing when I ask them to hover, I will be like “hold your mouse over the error for me so that it will pop up with whatever exception or whatever the problem is” and I will ask them like “what is that, why is that there “ kinda thing, So I am trying to lead the, so that next time they have it, they can look at the error message and be like “I have had this before”, that’s why I teach them the debugger too, because I’ll be like “oh we have a null pointer exception, drop a breakpoint on this line for the test you just failed, now go into the debugger and hold your mouse over the objects, and that one’s null, why is it null?” and they can go and find where it wasn’t assigned or it was never initialized so hopefully that will be modelling for them what the debugging process looks like, in the future, “Go look at the debugger, what’s the problem?”

S : I do that all the time too, a similar approach, where you are trying to tell them the debugger, where the first thing is where are you going to set the breakpoints, cause what might be the suspicious area or something to look out for right.. So first you just try to tell them “okay, now you did these 3-4 steps, before something broke down here, So it could possible be here or here or here, so what could be the suspicious point if I were to look, So I see that this could be one, and hence I am dropping a breakpoint here and not just like initially and then you like run through the next parts.. So that if they were to set their own breakpoints, they don’t go
randomly setting it anywhere were they like..It’s like they have a more clear idea of why they are doing it in some places and not the others.

Researcher :But not all problems have that scope right?

R I would say a lot of times, when I end up,... I think i end up modelling not on purpose but just I am trying to help them solve their problem so they are going to pick up on what i do which sometimes is a bad thing because, worst TA ever, sometimes I am too lazy to use the debugger I am too busy, so I just throw a print statement and I don’t know, I should formally go through the debugger, but sometimes I model bad practices, but at the same time, a lot of this like, we help them, we guide them with our leading questions, sometimes I’ll ask a student, “why is it”, what is the code doing here and they will say “I don’t know” and look at me like I know and I will say “ya, I don’t know either” (group laughs). So they think that I am asking them to lead them but really I am just asking them like “do you know what’s going on, I don’t either”. So a lot of times, that’s when I’ll end up modelling it with them, to work through it with them, is when they can’t tell me what’s going on and I certainly don’t know what’s going on, “okay, so let’s sit down and go through it together and we are going to go through”.But I get that a lot, when I ask a question, I legitimately want them to answer for me and they are just like “I don’t know. what is the answer” and I’m like, “I don’t know either.” (group laughs)

G : (adds)“I did not magically test your code” (R agrees)

A1 : Or for yes/ no questions, I get like “Yeeessssss ????? nooooooo ??? Maybeee ??” and i’m like “Awww”.

G:“yeah, they will just start guessing like, “thisss ??” and I’m like “no”.

A1 : If we asked them why, that’s when they will no, they haven’t guessed correctly..

G: I feel like drawing the pictures is trying to model it too.. If they really need to understand the structure ot it, they are clearly not getting it, then I’ll be like “okay, we have an empty list right now, we had one object, where is everything pointing? Okay now we are going to add the next one, where is everything pointing” so you know, you model the process..

A1 : sometimes I ask them to explain it to me in plain english, “what is the goal of this method?” or “what is the goal of this section of code?”. Then we are going to go back and say, so you have repeated.. I repeat what they say, “so okay, this is what you want to accomplish, let’s see if your code does what it’s supposed to do and find the mismatch ”

R : Yeah, I have a .. I remember that being a bigger deal in 1114 , asking someone what their code did, and then kind of repeating it to you just in code , like while this is not equal to this , and I am like, can you explain it to me in english, like you write it, let’s both understand it in english, I can read your code just fine, I want ot
G: Or they are just very confused that they don't say anything (R agrees).

Researcher: Completely off topic, what kind of advice or guidance from a mentor, if you were given a mentor, and this is the first time you are TAing what would you have, what would you think is good advice that you can get from a mentor, like when you started out, if you could have had a mentor, what do you think would have helped you?

G: I think it would have been nice to have some context about the curriculum here, cause I was just stumped in...

Researcher: Okay, any other things that would have helped? Or if you have to ask it the other way around, Now that you have the experience, what would you tell a new TA? Like watch out, these things are gonna come up or might help you to do this.

R: I would say it's really important to, it's really important that TAs remember and recognize, in intro level classes that they are not trying to weed out students, you are trying to help them through, and I think a lot of times, TAs don't realize how much of an impact they are going to have on their own students to get them to want to continue in the program, and sometimes you know they think like, “oh the students, they don't understand anything, they are so dumb, they will never make it, let's you know. I am not going to help them”, or or something, I don't know. But I think TAs can have a really big impact because that's the person you go to for all your dumb questions, you are not going to go to your professor, so reminding them you know, “there are no dumb questions” stuff like that you know. I don't know. I had a student, my very first semester as a TA, in my lab and he didn't understand anything but I try to stay very positive with him, like you know, he ended up failing the course, and I was like, oh my gosh, he is not gonna make it through the program and got very panicky about it, and then I left for a year, I left tech and then I came back and he had caught up to me, he was in all of the same classes, and now we are graduating together. So it's like, if someone had said, “you can't do this, you shouldn't be in CS”, he probably would have dropped out, but if someone tells him like maybe you just need a little extra time or a little help then, I don't know, I think it can make a difference.

A1: On that line of thought, also encouraging TAs to remember when they really didn't know things, I guess, for graduate students like us, we have been in different curriculums, we have been in this situation for a long time, remember when you don't know anything and when I was teaching 1114, a lot of that advice was of help, like “be careful with your pacing, make sure that you come with a lot of redundant examples, build up their confidence, ask them silly question, don't overwhelm them very much, try to keep things simple. And incremental.

G: And I would also make sure that people don't assume that they would understand the concepts always, if they come to with you questions, start by assuming that they don't
fully know what's going on in their own code, and they might need help, ask probing questions, before you actually start doing any coding.

Researcher: Any other thoughts comments?

S: I think most of it is covered.

Researcher: Okay.. I would also have personally. It would have been nice if someone told me that you should always think about the time when you were still trying to, like you said, “how was it for you when you first learned loops” and I was like “I dunno, I was told to do it, I did it, and eventually it became natural” but yah, it helps to think about when you started out..Okay... That’s all I needed, thank you so much.
Focus Group with Students - Transcript

Researcher: I am starting to record now. Feel free to answer these questions with, based on your experiences talking to TAs during your CS1114 or 2114 time. Any examples would really help, if you want to like quote an example. I am not particularly looking for any names, but you can just like share your experience.

Participant 1: Alright.

Researcher: Okay, so we are ready to get started. Okay. So when do you actually like decide, or at what point do you decide that you need help of a TA or what makes you go to a TA?

Participant 1: When I have been struggling with an issue for like 20 minutes or so, when I start to hit about an hour, looking at the same problem, like if I have hit a dead like a block, and usually if it's like an error that is reoccurring.

Researcher: Okay, has there been a time when this, it's not like you started and then had an issue but you didn't know like how to go about things? has that happened?

Participant 2: Yeah, so for me, usually that will happen during a lab, and if I don't know where to start, and I am not in lab, I will try couple of different routes, and see which one works best, and if I run into a problem, then I might talk to a TA but in the lab, because it is only 2 hours, if you don't know where to start, and you are running out of time, then it becomes a lot easier to just talk to a TA about it.

Researcher: But then you try to sort of figure out certain ways if that doesn't work, then you ask

Participant 2: Correct.

Researcher: How often would you say that you seek TA help?

Participant 1: Outside of a lab?

Researcher: It could be both ways, you can mention it as separate things?

Participant 1: Well TAs are usually my last resort... usually I can get through most of the problems, and then when it comes to web-CAT or something, about getting small points, and figuring out how web-CAT is parsing your code and trying to grade you, then you can go to the TAs and they could tell you what is really going on, or there is a lot of finicky things with web-CAT, so a lot of my questions are usually just about how to improve my web-CAT score, after I have gotten the majority of the problem done but definitely with, for 2114, it was a while ago, I think we were using jBlue? and the kangaroos,

Researcher: Do you mean BlueJ?
Participant 1: BlueJ, yeah, that was it.. I usually went to TAs when I've gone through the problem 6 or 7 times and the iterations for how the kangaroo would navigate the maze by itself, and I couldn't get it, but for that class in specific, there wasn't anything really crazy, so I went to them it was,basically to touch up my code and then look over my code and see if it was poorly written or not, to tell me how I should have written my code,after I have already written it..

Participant 2: so for 1114,I did not go to a TA at all, outside of labs.like you said, I mainly looked to my peers, would be like the first method, for getting help on work,and the projects weren't difficult enough,that I really like had no idea where to start,like I would want help from the TAs, but for the labs, regularly, almost every lab,I would ask the help of a TA,because there is some small quirks in web-CAT grading, that I mean if you had the time, if you are working on a project, you can slowly start to figure out, but if you are pressed for time, then you really just have to ask them for that and then for 2114,I went in probably 3 or 4 times, for the projects, and then just as regularly for the lab because the projects were a little bit harder,and there is nuances within the web-CAT grading that become more emphasized,in 2114 than in 1114, especially at the end of the semester when things just get really busy and you don't have the time to do the projects,as much as you would like to and then it just becomes easier to talk to a TA.

Researcher : Is there anything else that takes you to a TA? Maybe, like just something that was taught in lecture, or maybe grades,does that happen a lot or?

Participant 2: No..for me, If I was going to see a TA, It was because I had a very direct.. (Participant 1 intrudes) specific problem..issue with.. I went into one of the practice problem sessions with the TA , near the end of the semester, and that was somewhat useful, I found that for a good majority of the time it was to go in and ask them specific questions if you had them about the practice problems, They would be able to work through it together kinda procedure So I spent a good majority of the time there trying to work through them myself as opposed to Doing like a true collaboration with them..

Participant 1 : Definitely for labs the 2 hour time limit for certain things like learning binary trees recursively , It's a lot easier, like the 2 hours put A lot of.. pretty big time constraint on you because some of the concepts are pretty difficult especially when they give you a block of code and tell you to fill it out where they have missed it and so you read all their code And then try to figure out what you're supposed to type in there to complete it. To figure that on your own I don't think that the two hours time limit is correct for that Because I think when most kids go into that Actually just look in the slides for 2114 are basically she gives example code Or she tells you directly how to do the lab so it's not like you read it and figure it out and reiterate your own problems to figure out eventually.You need to just ask the TA I look at the slides that she gives you to really get a good foundation of what you're supposed to be putting in those blocks And then start writing your code because to do it from scratch And see how you need to iterate out If you are removing Some sort of a node with two children. It's kind of difficult to do word from their code
Because it's not like you have built it all up. So they're looking for something specific

Participant 2: Yeah I agree I would say in the labs it would be a process of seeking different routes. The start of the problem and to code up something and then an hour and half in, you realize that you are further entrenched in the issue then you would like to be. You haven't gotten anywhere and so then it's essentially like trying things that fail until you ask the TA and they help you till submission.

Participant 1: You can't really dynamically solve the problem when they have given you a lot of the foundation of the code if you are not interacting with the code correctly. It's not like you have built everything from scratch. So when you run into a problem you can't go like “Oh I'll just add another line here. Another thing here”. You're supposed to be working with what they have so and that's beyond 2 hours so if you start just basically recreating the entire code for yourself.

Researcher: But do you think that one and a half hours of failing actually teaches you anything about how the code does not work?

Participant 2: I think (Participant 1 agrees too) Yeah I think it definitely does. I think it helps you to better understand why when the TA shows you the solution that is the solution. Because all the routes you have tried before have failed.

Researcher: and you see why that's failed when someone tells you why something else works???

Participant 2: Correct. Having said that the point was that I haven't found the procedure in the lab to be continuously failing until you find the right route. It's continuously failing until the TA shows you the right route.

Participant 1: I can see that... Some of the problems can seem really daunting if you were to just give them the prompt and write it yourself and if you were to do something that was not recursively, which is actually iterative, you can see the recursive method, when they give you a couple of parts and another example which you can base your code off of, you can see how elegant their solution is and it is like 3 lines of code whereas if you were to do an iterative method on your own, it would be 50 or so...

Researcher: yeah.. yeah.. Okay. Have you ever stopped yourself from going to a TA for help? Like if there is an example of such an instance, I would like to hear that...

Participant 1: Not for CS, but I think it's a general thing, across the campus. If you know that you can solve the problem by yourself after 2 hours or so but you could get it to a TA and get it done in 20 minutes. If you're doing a project well enough in time you might just save yourself the time and it'll be really quick instead of you have hit an error and just changing your code or a large block of your code to avoid that error completely, because you are not... passing a variable or you don't have access to a certain variable, you have to change how you do a process. But I know in EE or these classes, you can sit in a queue or wait for a TA for 2 hours if it's sometime you know
it's crunch time, 2 days before a test or a few days before a project is due, and it's usually better, you will learn more if you do it all yourself.
It's not going to be any quicker, if you go there and wait 2 hours to get , you know, they are trying to go to everyone because they have so much demand on them so the answer to your question is probably going to be very sparse. They're just going to tell you what is wrong and give you a certain example to go on which might not even help you that much..

Researcher: and then you waited 2 hours for that?..?

Participant 1: yeah,..So it's better to tackle the problem yourself in certain cases..

Researcher: mhm.. Any other case where.. maybe the TA was bad or they are not going to help you, you know that.. maybe the kind of problem you are stuck with is the kind that they will not help you with ? I don't know..

Participant 1: I don't wanna be mean or pick anyone out but for EE in specific, all the TAs are indian and some of them have very thick accents and so it can be very difficult to get help from them , because those are such.. having a technical barrier, on top of a language barrier can make it very difficult. Although they understand the problem, it's hard to communicate it in a way that you will start to understand it and with these technical fields, CS and everything, it's very hard to communicate. so.. Certain TAs can be better, especially I am sure if there's a student who can speak in indian, however you wanna communicate, would be easier, but for EE there are certain TAs that are easier for m to communicate with..

Participant 2: For me, The issues that I've had with TAing in the past , In 1114 and 2114 is that when... so best case scenario ,When you go in you talk to a TA ..you wait in a line for a long time and you go, finally get seen with them and then they will show you how to do it, and they will leave. You are with them maybe 5 minutes.. And worst case scenario is you go wait and wait and then they're done and they leave. And so in the best-case scenario you are getting the answers from them They don't have the time to work through why that works , with you .So I can a lot more of that in the lab , they are actually able to go through like “this is why this happens” Because they are there for that full two hours. Because everyone is trying to get word with them in those office hours with them like I said before you get the answer but you really don't have a good comprehension on why that's the answer or how it works and so , a lot of the time for me , it's been a pro and con versus , Do I have the time to work through the problem myself, because it would be more meaningful and rewarding So me id if I got the answer myself versus Do I have the time to go in and wait and talk to a TA, because it's more of a guarantee that I will get the solution and less of a guarantee that I will understand why that solution works.

Participant 1: for 1114 and 2114, If you are given a prompt and you can solve it however you want, It is going to be hard for a TA to look at it and understand it, you have gone some weird path that is not the most efficient way , It is just how you have decided to go about solving the problem, they are probably going to give you an answer like, unless it's some minimal problem, they are going to say “this is probably not how
you should do it, you should probably be going about it a completely different way, which would avoid your problem completely’ if you are running out of memory or for whatever reason, your memory management is terrible for a simple problem, to have a TA kinda shut you down and just be like, “yeah you should probably scrap your entire code, the best way to solve this was completely different than how you are doing it” tell you to do it recursively, if you have been trying to do it iteratively, or something else, which can be a. I mean that’s a part of the learning process but that can be tough because, TAs answer is going to be “I can’t help you, you are going to have to do it a different way, to solve this problem” but I mean that just goes with technical studies.

Participant 2: Yeah, that’s what I was going to say, It’s just kinda one of the inherent difficulties with programming, is that debugging especially so elusive, so you come in and say, this if statement is wrong, and the TA looks at it, and it’s not the if statement that is wrong, and then the TA, at that point, it’s either, “Well, I could spend a while looking for whatever else in your massive code is wrong, knowing that this student probably implemented a lot of these things inefficiently or I can tell them what’s right and move on to the next person”,

Participant 1: That doesn’t help at all the queue times, if they try to actually help them they will spend 20 minutes on a student, that’s too much if there are like 7 people waiting but I mean that’s manpower, you can only have so many TAs, there is a ton of classes.

Participant 2: There is also, I mean I don’t know if there is a more efficient way to do this, but TA office hours are just too hectic to begin with, and there is no line system or anything, it’s just like whoever is the most aggressive with their question will get seen, and so like, there are kids who will be talking to the TA and then they will be answering the question, and be like “what about this one” and so they will have like 8 of their questions answered, before the entire rest of the line is even gotten a single word in.

Participant 1: They kinda do it like a free-for-all in the CS lounge. There is no Queue. It’s not like you sign up to be, for them for 20 mins, and then they will reserve 20 minutes for you and then the next person will come, there is nothing like that to bring the example back, for the EE labs, they have a queue and you just type into it a tinyURL and go there and sign up and then they will spend a certain amount of time with you and move on to the next person which kinda helps specific persons, but if you have the same question over and over, it’s probably not the best use of time for the TAs.

Participant 2: and I mean, for me there have been times, when I have gone in and not known where to find the TAs that I was looking for... Everyone is just sitting there with their computers, so like who do you ask, who do you talk to because it is not blatantly obvious who the TA is...

Researcher: we do have those little orange placards?
Participant 2: They do have it, but if it’s at the center of the table, and there are 8 different kids, and you are trying to figure out who looks the oldest, you know what I mean...

Participant 1: I think it would be best if they got on the game for the canvas profile picture, some of the GTAs fill it out, makes it a lot easier, to a point if they don’t have placard or if it’s between the table, you just have to resort to shouting out their name which can be difficult, without butchering it.. it’s a foreign name..

Participant 2: It’s one of those things... It’s like if you are not an outgoing person, then you would have a very tough time of seeking out the TA once you actually get there and getting the help...

Participant 1: the best way is to just ask someone..if they have seen them around..but even that.. going around like “have you seen this TA?” “I don’t know that person” and you go around till someone goes like “oh yeah , she is not here today” or “he is not here today”.

Researcher: I think we touched on this, but what is the kinda help that you expect from a TA? and does it depend on different things or just that part of the question like, what kinda help are you expecting from a TA?

Participant 1: well for now, when you have the queues, the way that you actually have to go about it, just for, basically administrative purposes, how many kids you have, they will just kinda tell you the way that you should be doing it and maybe show you some example code, or verbally tell you what your code should look like, and that’s probably the best learning process, in the ideal scenario where every kid ad a GTA, they are really involved where the read the code and tell you what you are doing wrong, personally I prefer them, if you are going to a GTA, they should give you more help than you can get by googling..there is no point going if they are just going to tell you like “oh this error is usually because of input error” ut I googled the error yeah, I know it’s for an input error, that doesn’t help me at all. So it has to be more involved, with them looking at your code and see where the error is, nd maybe not tell you directly but tell you to look over a portion of your code, and then you can go out and look for it and if you know you have narrowed it down to one class, then you personally can look through it and figure out the error on your own. Just telling you what the error is, that’s no help there.

Participant 2: and I, I mean, it’s one of the just difficulties with TAing computer science, specifically is that, computer science is very complex, and that you can get an error that can be rooted in a billion different things, that’s causing it, so like at the end of the day, it’s impossible for a GTA to have all that information memorised, to know exactly what causing the issue, at the same time it’s hard to not feel like your time is being wasted when they start googling the issue as it comes up, at the same time, I should not expect them to know just off the bat, like “oh that’s caused by this if statement”, I dunno, it’s difficult, because you expect the GTA to have all the answers, so when they don’t it is kinda.. you know.
Researcher: Is this any different from a UTA, because I was just talking about TAs in general, it doesn’t have to be a GTA, so if you guy’s note a difference between GTAs and UTAs, please talk about it.

Participant 2: I mean in my lab I will generally go to the GTA for help...

Researcher: Is there a specific reason to that?

Participant 2: So in the past times, I have gone to UTAs, if the usually doesn’t know the answer, they ask the GTA and so people catch up pretty quickly that if you have a problem, then there is a better chance that the GTA has the answer to it.

Participant 1: Also, the UTAs are a little bit more disconnected, they are in higher level courses so they haven taken this class in a very long time, most of the times I believe, so the GTAs have taken the class in the past, 2 years-3 years or something like that.

Researcher: Well you could think about it the other way too right? I mean, I am not suggesting anything, but we actually don’t, I mean we are not introduced to Java they way you are, like in India, it was very different, but then yes, like you said I have TAd this course multiple times before and I kinda know the usual pattern of questions that come up, that is true, but I would think that is the same of the UTA as well because they have also TAd the courses before, and they have also seen it first hand sometimes, I do know that some people do prefer talking to the UTA, because UTA can maybe relate to them more like “I have been in this situation before”.

Participant 1: Okay, like they can see it from your perspective?

Researcher: Because they took the same course in the same campus.

Participant 1: Which helps... it helps... because I think if you haven’t taken the course, even, I mean if you have taken the course multiple times, you have seen it, TAd for it a couple of times, then there really isn’t that much of a difference, because you have seen the course load, you have seen what it’s like, but if it’s your first time or second time, I mean you are solving it from a very general perspective, you know Java and you know the problem, you are just going to solve it that way, when there could be a, although less efficient, more direct way of solving it.

Researcher: So are there any factors that go into the kind of help that you expect, is it like “Oh I have to submit this tonight, so maybe I am not looking to understand it as much as get it in?“ does that happen?

Participant 2: Yeah, definitely, I mean, a good majority of when I go see the TA is when I am time crunched during a lab or during a project, and at that time, of course I would like to understand it, because when the test comes up, I would like to know the material but, when it gets later and later and you are getting more and more frustrated then obviously you would like to have the solution.
Participant 1: Yeah, you kinda want to just push through all the errors if you are there, if you are on a time crunch and you hit a block, you just want it to be gone, to come in and tell you what is wrong so you can change it and continue coding until you hit the next block, which is kinda... it is really inefficient, because basically you are just there and you are writing up, you hit an error, you ask them, they solve it, you go, you get another error, you ask them. You are just constantly going into the pool of asking questions, because I mean you have procrastinated too much, obviously, but that is really just procrastinating, just no way you can force kids to, unless you have pre-submissions and things like that. It’s going to be difficult to get rid of that... but yeah.

Participant 2: and I mean, maybe this is not playing off the question too much, but for one of the projects in 2114, the roller coaster one, project 4, and I was getting an error that I did not understand, when I submitted to Web-CAT. So I went in and it turns out I was getting points off because, so it prints out the height of each person in centimeters, and so the person is 80 cms, I was printing out 80 space cms, and they way they wanted it was without the space, but you never would have known that, if you had not come in, and apparently, the TA was saying that tons of kids had come in with the same problem, that was just one of those things where you never would have gotten it if you had not gone and seen the TA, so sometimes, instantly you have the answer, it is not necessarily like, trying to jump through the comprehension because you are on a crunch for time, but because it is one of those things where you can’t work through in your mind, it’s not a reason thing, it’s just a technicality, that you wouldn’t have if you didn’t go and talk to them directly.

Participant 1: that’s one of the issues with web-CAT grading, because there are plusses and minuses too, having you have been completely graded off of just a static output to a prompt, the command prompt, which can be, you can miss something small like a space and have it, basically it will give you full errors, and you will get zero percent because you did not have a space in there and it just everything is lined up incorrectly, but it also is good because I mean, if you can follow their prompt exactly, it’s not testing you on your code and if your code worked it’s just the output, the final result, it can be nice for somethings, if you code it up in way that’s hard to write test examples for and things like that.

Participant 2: I missed a lab this semester and I reached out to my GTA, and she helped me make it up and then I was not able to meet for the full time I was supposed to make it up, so she reopened submission for an hour, the next day, and I just thought that was super super helpful, I think that made me a lot more comfortable reaching out to them, not seeing them as this, like, I don’t know, this distant figure, but someone whom is actually willing to like help you through more so, than I would think, the professor, because I think because the GTAs and the UTAs are in charge of smaller sections, then they are able to devote more time to them and where like if I had an issue with my grading, I would feel more... I don’t know if I would feel more comfortable talking to them directly about it, but I would feel that I would have a better chance, at my GTA or my UTA helping me thought like if I ran, if I got a zero on something and I wasn’t sure, I would feel much more comfortable going to them than like the umm, the professor.
Participant 1: Isn't the UTAs who have access to opening grades and opening thingss that are beyond just labs ??

Researcher: I am not sure about 2114 but in 1114, so.. it works a little differently, in 1114 we open labs on Monday, the professor does that and they don’t actually have to come to lab to do it. It is even from Monday, they just changed it this semester, and GTAs grade the programs, UTAs used to grade homeworks but then that has also changed this semester, so I guess that depends on how the course is set up that particular semester.

Participant 1: yeah, because I know for.. I think UTAs for 2114, I mainly go to them for administrative things, when I need them to help me open a grade or if I have gotten randomly a 0 for something I have submitted, just to see what went wrong between my submission.. because on web-CAT you can see what your grade is and the TAs usually have to grade a certain portion of it, and so you can see what went wrong, when people are actually grading those assignments but for EE for TAs, you submit a program, there is no web-CAT, it's you made sure it worked on your microcontroller and they upload it, run it on their microcontroller, there is just a massive disconnect where if you set up something wrong, you just got a zero on the assignment, which is worth a lot of your grade so you have to go to the people who are in charge of, or have the ability to change the grade, and have a little bit of autonomy from the teacher, otherwise you have to, you got a GTA and then you just end up emailing whoever is teaching the class and to email them is a real pain because it just feels bad to email them a bunch of like text questions basically because I am sure that they have a lot of kids to administer and multiple classes perhaps and so filling up their email is, it's just.. there is going to be a lag in response, but for teachers, I know our professor is on top of her emails, a lot of these teachers are really on top of their emails, and just get back to you so quickly, but I would rather talk to a GA hen they have time in their office hours because that's you know, they have blocked out that time for you.

Researcher: okay.. Do you prefer to go to a particular TA or a set of TAs? like, if you had the option and there were 2 TAs sitting right there would you prefer one over the other or ?

Participant 1: Yeah, definitely, for certain classes, I said like I don’t think a lot of american students go for an overgrad, there are a lot of foreign over-grads that are TAs, I know some of them are just for.. for my microcontroller class each class has it’s own specific TAs but you can go to any of them, but some of them are a lot better at communicating than others so, I know for bowman's GTA is really really good, so I prefer to go to him when I have a question because he knows how to solve the problem pretty quickly and he has been doing it for a while, so he has more experience with the problem people see.

Participant 2: For me, so I have a GTA in my lab, and I have gotten some really weird errors like bugs in my code, and she will be like working through it and she will be like chuckling like “I don’t know why this is happening” but even if like she doesn’t know why it’s happening like even just that little bit she will establish like a
rapport, I now feel much more comfortable going to her more than a random TA, just because like I don’t know, she has worked I think like, she establishes a small connection with the people she works with. It makes it easier.

Researcher: So it is not uncommon for you to prefer your own TA over other TAs who..

Participant 2: That is correct.

Participant 1: If you build a relationship, you don’t want to be criticised, I have never had that happen to me but just show someone your code and let them look at it and go “wow, this is just terrible”, “this is an awful way to do it”, there are different jokes I mean, “just get past web-CAT”.. You build a relationship with certain GTAs because I mean things can look really really ugly, going from web-CAT, you have basically blocked out all your if statements to be cascading if statements because you don’t want to AND them together because it makes just testing your code much harder, because it has to go through every single possibility but yeah, if you build a relationship with one GTA I mainly go to the same one, like for each class every single time, unless they are not available..

Researcher: But has there been a case where you found that a different TA, the way they make you solve the problem like.. has that happened?

Participant 2: Yes so I have had TAs that in this class specifically, I have had TAs that will be more willing to help you work through it, and then TAs that will just give it to you directly and part of that may just be symptomatic of the point of the semester, like where in the project you go to them, so like, I went talked to a TA, I remember in 1114, it was like a couple of days before the project was due, so it was not like the worst time crunch, and they just took my laptop and typed up whatever I should have had.. but then 2114 this semester, I ended up turning in a project a day late, so I went in the day after it was due, I guess the evening of the morning it was due, and I went into the lab and the TA was essentially like staying there with me and working me through it and helping explain to me just because like it was after the massive flood of students trying to get it in on time, so they weren’t that pressed, so that was a lot nicer.

Participant 1: it varies on how much help you get, for labs, for electronics, they will just tell you “oh you should have this block of code here, this is how you interact with the external device” and some of them will just be like “read the data sheet”. so yeah, some TAs you want to go to because they give you a lot more help, sometimes even though they are giving you the answer basically, being able to see the answer, I don’t wanna say monkey see monkey do, but once you see it done once, you can usually apply to other things, even though maybe struggling and reading the data sheet, putting out a whole bunch of different types of code whatever eventually you get it and you will learn a little bit more, the amount of time you save, if you re-invest into learning is better..

Researcher: How is getting help or learning from a TA different than learning in lecture or getting help from peers? or any other mode of getting help?
Participant 1: 2114 lecture is awful. I mean lecture is to like a 500 students, and so they usually, I mean it depends on the teacher, you can have a teacher that goes way too fast or a teacher that goes really really slow, and for 2114, I think a lot of the introductory coding classes, They like to set up a kind of slow ace, so they can assure that everyone is kinda up to speed on everything, but If you have taken a couple of coding classes before hand sitting in that lecture is almost a total waste of time, they will pick up on a certain thing, you will see an example code and that will help you, but a lot of it is just meaningless, if you go to a GTA all the stuff you already know, you already know, you can ask them about specific things like you can cut down on a lot of the waste and especially for [Professor A] you have to come to class because she does in-class tests or quizzes, so there is no.. you go because of the quizzes, you don't go because the material is something that you need to pick up on, which can be.. I mean if the class is going really fast, I mean that’s a worst case scenario than going too slow and everyone is trying to catch up and no one can and then it’s more trouble on the GTAs but then if you are going too slow, for people who have coded a lot it’s a little annoying.

Participant 2: Yeah, I mean the distinction between sitting in lecture or working with the TA Give me is just based on what you're actually doing you know what I mean I mean when you're sitting in the lecture you see like a snippet of code ,You will need this in this circumstance and okay you put that in the back pocket but you're not actually using it, you're not doing anything with it You’re seeing it up on a slide and “I believe you “, But you don't actually like try it.And so it's just talking at you Which is exactly what a lecture is because programming is So learn by doing So it's very difficult to get the same amount of meaning from a lecture As opposed to when you're sitting working on a problem you are entrenched in and you are waiting through all the difficulties , you're frustrated you getting your hands dirty And so you're actually learning by doing , and then when you are working with the TA and even If they do give you the answer you would have attempted it So it's more ingrained in your memory at that point than if somebody were to throw something new on a slide and it just has no context in your brain because you’ve never used it before.

Participant 1: Yeah I think I can second that because the learn by Example I'm going and writing the code, Can help a lot because when you are in lecture and she shows you these are the outputs, this is what will usually happen” or She just gives you the function that will output or the algorithm It uses and then gives you a question on it, It's hard to just write that code on a piece of paper And always get the right answer, it's better if you can code up the example yourself and throw out a couple of inputs into it , see the output ,do it again a couple of times And then you start to really see by example what the outputs are,Instead of just trying to follow an algorithm in your head Give the answer without using any code at all. I think that also a big problem with the lab , the 2 hour time crunch . because they usually give you pre-lab thing which is like writing out the test Examples it really doesn't help you with writing the algorithms That the functions work off of And so when you get to that you feel like “I haven't done this at all before”, Sometimes it comes really quick because you went down the right path And the labs over pretty quickly and you're done But if you went down the wrong path You dig yourself into a rabbit hole. You come out
an hour later and then you are like basically have to scrap the code because you went the wrong way which makes the time crunch even worse But luckily the penalties for going over time like that, That is like 5% or 10% which isn't crazy but it seems kind of odd to put you in a situation That's like "you have to learn this in this tie, that doesn't really take into account, the kind of variations that people have.

Researcher : mmm. How is it different from learning from peers, when you get help maybe on piazza, or maybe when you like sit and discuss with somebody, how is that different from learning from a TA or sitting in lecture?

Participant 1 : Depending on the assignment, peers are more free to tell you how they did it, they could have taken the same path as you did, but they can tell you verbally the code that they went through, so it's a lot more specific, not as much on the technical aspects of that the error means and "Oh I solved the error by doing this", by putting in this line of code" so it is kind of ask reciev kind of thing, where you if you hit a problem, you ask, if they are farther along in the assignment than you are, they can just kinda tell you what the answer is, which for some assignments you are not supposed to do that but beyond that, peers are usually, you can just facebook chat them, if they are your peers you can talk to them through text, you can't really do that with a GTA, peers don't have a line of people, asking them questions, so usually, if you have a couple of peers, you usually all kind of take rotations, usually you're not all procrastinating, so someone has done the assignment before everyone else like "ah, yeah, I had the weekend so I did it" and so you can ask them or vice versa. So sometimes you will do the assignment, if they ask you a question you can help them out, so It's a lot more personal and it's easier way to kind of force you through an assignment if you are on a time crunch, because there is no line, you are sending them an impersonal message facebook or text, you can't really do that with a GTA, that doesn't make sense for them, they can't open a facebook page and you just start asking questions through Q and A.

Researcher : but the kinda learning, like on an online forum, maybe you didn't ask a question but someone else did, and the learning that you get from that, or the answers that were posted or ?

Participant 1b : what was it called, stack something ? I forget.. or CS stack, it’s one of the main forms in which they have all the different questions posted about general iterations and how to make it an iterative class etc, so you can just look at their examples which I think, because usually if you go to the actual, they don't call them datasheets, the oracle website with all the java examples and the subclasses, they don't give example code, so they tell you what the thing is and what input it needs, and its outputs but sometimes you need an example code just to see how they structure it.

Participant 2 : Piazza I found to be a luck of the draw kinda thing, I mean nobody uses Piazza as general education, nobody builds on it just wanting to review, usually if they have a project and they have this very specific project and they don't want to wait in the really long lines for the TAs, but generally the only time that other people will be on Piazza, if the professor isn't dispersing the question is if another
student has a very similar problem in a very similar window of time and so it’s really just like you post something on there, and you pray that had a similar problem and solved it and is also on Piazza for that problem or a different problem and so will see it.

Researcher : More Like being in the right place at the right time ?

Participant 2 : Exactly.. so I have had luck once, with Piazza, getting an answer to the problem, but It is definitely not my first route when I am having an issue.

Participant 1 : I think there , you can have certain classes where there is basically piazza student heros where if you post a question, and you can answer questions as well , you can basically go through the assignments as a group, I mean in 2114, there is just random things like, really niche little problems that a lot of people are having to have the right outputs, I forget what the ast project in 1114 was, but I remember on piazza, lot of people are having the same issue, and It was just initializing something.

Researcher : the rocket thing?

Participant 2 : hm.. I think it was.. 1114 was the asteroid game.

Participant 1 : It was the asteroid games, I mean you have to work within their thing and they sort the objects themselves, I mean, you don’t know how that works behind the scenes, you just use their calls, and I think there were some people who had some weird collision errors. and you had to set up your collision right and set up your objects right to have correct collision, and I remember there is like 6 or 7 questions, and then eventually there was one student who said do this, it was like 2 lines of code and then your collisions will work correctly, which.. yeah.. Piazza is, is kind of give or take, sometimes I think, If you don’t procrastinate and you spend the time on Piazza, I think you will get a lot out of it, because I think a lot of other students will... personally when I am done with something, I look through Piazza and see if I can answer one of those questions. And I go pretty in-depth, I try to put a picture or something up but yeah, it depends on the class, sometimes Piazza can just be like the answer will just be directly given to you, sometimes.. I know for the collision thing, when I was looking at it on a weekend, there was a bunch of questions but there was no answer..

Participant 2 : Yeah, that’s true, I feel like people will be posting intermittently as they working through which obviously is different timeframes for different people, so maybe somebody wants to get a headstart like 2 weeks, so they have posted a couple of questions on there, and then nobody will say anything, and then maybe a couple a days later, another person has nothing and it’s really only like a day or two before when everybody is going through it for their specific problem and all the ones get answered.

Researcher : Has it ever been the case that you saw a post on Piazza and go lie “ why did I not think of that? “ you know, has that happened like maybe because those things go on record, as opposed to you talking to a TA , you know what I mean ?
Participant 2: "I didn't think of that " in what sense ? like way to implement specific piece of code?

Researcher: maybe.. that was just an example I thought of like right now..

Participant 1: I think like, if you can see someone else's code that they have posted, I think a lot of prior experience can make a certain problem very easy, like a lot of people in 2114 don’t understand the reason for having a main a parent class, having a whole bunch of different ways to measuring a rectangle, circle, and be able to put them all under one object. But if you can put them all under one object, you had experience doing like a game or some other thing, you can throw them all under one iterator and then you can iterate through all of them together, you don't have to a whole bunch of different iterators for every single different class, and that's the same thing like with the asteroid game, you can put all your objects under game picture or game gif or whatever it is for you asteroids and your little layer being the same thing, but you can iterate and update everything together, it's not as hectic, so if you can see how someone else has done that, because if you are managing like your asteroid or your different asteroid types, not iterating together, the code would just explode, I can understand, once you have so much code, it just becomes really difficult to debug, because you are looking through so much, you can't really share code but seeing how other people have done it, set up their classes can really help, or just, it's kinda hard to point you the right way to see how you have to do that, it's kinda specific, you understand objects and classes and this is how you would iterate through them together, is not something you usually go over, it's just there is object classes and how they work together, not really their application.

Participant 2: It's also.. I am a pretty visual learner, as opposed to an auditory learner, and so If I finally get seen by a TA, they throw me a hint, and then they leave, and they are talking to somebody else for 20 mins, then I will like start working on it and then by the end, I will be trying to remember exactly the nuances of the hint that they gave me, whereas if it were printed out and submitted onto a forum , I can re-read it like every now and again, If I am like still getting the same error to say like “oh maybe this is what the problem with this is..” as opposed to just hearing it once, and then expecting to have that guy with me for a long period of time debugging..

Participant 1: There is nothing worse than sitting in a queue for like half an hour, getting a hint, and then them leaving because they thought that the question was something specific that everyone is having, and then have that hint almost not apply to you at all, or you fix it and then that’s not the error, and so you have to sit back in a queue, for another half an hour to have them come back to you and ask a similar question which they thought that they already solved, it will probably peeve them and peeve you together so you will both be not so happy that you have to work with each other.

Participant 2: I also think that because Piazza is like documented in that, once you post something it's on there that people are more methodical with their hints, in like
helping people along and so you could get like some casual offhand hint from a TA that doesn’t help you at all, whereas I feel like people have to like really, they have to compose if they are trying to post on the forum, so you think about the best way to teach it to somebody, as opposed to like this is wrong.

Participant 1: usually it’s like on Piazza, you have a response feed underneath of it, so even if there is a couple of different variations of the problem and how to fix it, people will post if it didn’t work for them and what did work for them ad you can usually have ten or so responses underneath of the questions, solving it a couple of different ways and one main solution changed a couple of different ways, so yeah... it’s more tailored to different types of problems.

Researcher: which do you think is more effective in that sense? like in terms of learning then? talking to a TA in person or using help here or going it through by yourself? or talking to peers?

Participant 1: TAs is really nice if you are going to them early, if you have them to yourself, and they can just sit there and help you, then you can just pull them in basically for as much as you need and then you can solve the problem yourself, and then if you have hit something that you really can’t push through for whatever reason, they can help you and then they have the time to actually read, but when you are on a time crunch, there is, I find it really hard to justify even going, when I know that there is just going to be like 40 people waiting for a TA, they just can’t spend the time, to actually go through one person, it’s justifiable, I mean I get annoyed if I see there is 40 people waiting and I see some person that really menial, like little tiny problems that are, they should be able to solve, but I mean, I Don’t want to be, I mean everybody has problems in their own way, so even if I see someone like “oh that’s problems easy, why are the wasting time, I need help“ it is kinda selfish but yeah there is almost no point going to them, cause that’s too much time to wait for a little tiny piece of time when you could probably just spend the tie looking at the problem yourself and figure it out.

Researcher: Have you ever felt encouraged or motivated after receiving help from a TA? has any TA actually made you feel like wow, you can do this yourself, you are the code hero, you got this?

Participant 1: I like it a lot when.. I have spent a lot of time on a code, I have spent a significant amount of time trying to make something, not just work but also be efficient and they look at it and “oh wow, you have actually spent some time...” they can actually understand my code and that’s very nicely laid out, that’s motivating to keep doing that, keep on putting in the extra effort to really make your code look nice and actually work efficiently instead of having all these calls to different things everywhere that you probably shouldn’t have, that’s really nice. So when they can look and fix your error really nice and easily because they can see the error that’s ot as jumbled up and if statements everywhere, that is motivating.

Researcher: That kinda feedback and the fact that they realize that you put that amount of work into it?
Participant 1: Yeah, I have never had a TA look at my code and be like “wow, this is awful code” because, I mean, I know for certain things like the roller coaster project from 2114, my window was a mess, I did it from procrastinating too much and my window was awful and so I feel bad for the TA that went through that because halfway through my code I realized that their was a call that I could have used to iterate through the window so I just started using it halfway through when I wasn't using it for the other half... so there is some niche calls like they don't provide like an index thing so it just go through however you added it so I had to put in some like really weird variables to make my own index, so if something had a width of like 15, then that was a specific item, which doesn't make any sense at all, you can't read through that and understand that, so them knowing how I pulled the roller coaster object wouldn't make any sense to them at all, but yeah, I have never had them just look at it and go like “you need to scrap everything, this is awful " but usually this is in labs where you have looked at what they want you to do and so you can get it on the line, you can just tweak what they have but it can be difficult.

Participant 2: Yeah, I wouldn't say I have ever been particularly encouraged or discouraged either way by a TA, I mean usually when you are going in there and it is during their regular hours so you are crunches for time and they are crunches for time and in the rare case where they have time to sit with you and talk with you through it, then survival is the name of the game and then you get it until it works, and you are working through it until it is fixed and then when it is fixed you leave, you know, their is no exchange of like you know, “you have improved” or “you have done well” at the same time like “that was crap, you really didn't know what you were doing”, It is just a lot of like how do we get you there and then move on to the next person.

Researcher: So really no time for establishing a..?

Participant 2: Yeah, exactly

Participant 1: I almost wish.. it would probably be more of a curse than be beneficial that the TAs could go in and solve certain test case errors for testing large if-blocks because there is nothing more annoying than almost depressing when you have almost a 100 and you are missing 3 points or so because you haven't tested this class completely in every single situation and you have spent 30 minutes just trying to get that niche example that gets that 1 an I am pretty sure with web-CAT, they can be impossible cases to test but it will still hit you on so you have to do some sort of if splitting, but yeah, I mean it's usually just good to know that the TA can relate and say “yeah, it's a pain in the butt to work with web-CAT “ because they also have to do it most of the time, for whatever class they are using web-CAT for, just to know that you are alone in the fact that some of this testing is, adds on extra time to the project, but even if you have everything right and your code looks good, that you have to test every single situation which happens, which will be eas for somethings and almost impossible for others.

Participant 2: Yeah I mean the case that I was talking about was, like I said during normal hours people are busy you're busy they're busy, and then you get the rare cases
when You are going in and they Have a little more time to talk with you and they will relate a little bit more to the problem you are having and then you just kinda get the reinforcement like..

Researcher: it happens to everybody?

Participant 2: that this is just hard, for everybody, you know, everybody has to struggle through it, so it’s not necessarily you should be discouraged if you are struggling, because that is just something everybody goes through. (Participant 1 agrees)

Researcher: do the feedback that they write when they grade, doe that like or is it also very you know, problem specific or?

Participant 2: yeah.. yea.. I would say, at least for the problems I have had, it is specific to, like, I guess, like I could be seen as a summation of problems to a TA, which is a weird thing to say... but like if they are grading the project that I am doing hen the feedback is like maybe fix this change this, and at the same time, there is no like “you did it, congratulations”..

Researcher: do you think that would help? I mean, would it be nice to see that?

Participant 1: I think that would feel hollow..

Researcher: really?

Participant 1: I mean just to have that in text, “good work!” In person, it would mean more..

Participant 2: I would appreciate it..

Participant 1: I mean, if they have to grade a whole bunch of things, if they told certain things where, because for 2114, because if you submit something I have lost points for forgetting to put something in my UML, just some random classes like I forgot to put the node class for it, node iterating list, which is like, obviously the node iterating list would not work without the node class, I know that, I just, it’s not in there, but I will lose points because obviously it needs to be, and so there is not much text on that, just “you need to have the node in here”, It’s like “oh shoot.. “ so having “oh you did not put the node in there, but otherwise good work”, It wouldn’t be hollow, but I don’t think that will make me like “yess.!!” I don’t think it would spur me on..

Researcher: I guess I was just thinking about when I grade, I am like, when I actually don’t have any comments to give, and I am like “wow, that’s a 40 on 40, good job..!!”. because I am usually picking points off for not writing proper comments, and this time everything i in there, it feels like all my previous comments have been taken..
Participant 1: It is also unfortunate, the way the comments work is usually, if I get a good enough grade, I am not gonna dispute anything the GTA says, if they took off something small it’s not really worth the time trying to get 2 points, personally for me, it’s not worth that, I am not going to bug the TA or bug the teacher, try to get a few points, so for me I usually don’t even look at the comments, If I see the grade and I am like “I think that’s what I should have gotten” but If I see a grade that is really really bad for whatever reason the I will read through it and look and, make sure I understand this but the positive reinforcement I am usually not looking at it, so I am usually only focussed on the negative ones, so when I Get it, I look through all the negative comments, and go like this is what I need to improve on.

Researcher: but maybe more in person? if that happened in person?

Participant 1: yeah, I would prefer it in person, than just reading, I mean usually it’s not like “ow, this code is bad, I am taking off 2 points for efficiency, It’s usually like.. uhhh.. you have made an error somewhere, that you just really need to fix, like you are not handling a special case, so it’s like “yeah ok, I understand why those points were taken off” and especially for one project I did that, I can understand everything she took points off for, it was awful code, I can look back and I don’t really, that’s unfortunate that that’s tagged to my name..cause it was awful code.

Participant 2: Yeah, I guess I have just come to expect that from Computer science because it’s very technical and I know very little compared to a lot of other people in the world and so, If they are making a comment about something that I did wrong, It is just how it is.

Researcher: But I guess not everybody takes it that way some, I have personally seen that, in 114 people go like “I should just change my major.. this is not my thing..I cannot code “, and I am like “ that’s not it, this happens, even like sometimes I look at code and go like what the hell is happening here like, nothing makes sense, it happens..” so sometimes that helps.

Participant 1: I think it can be really difficult, depending on which high school you went to and if you did any coding yourself cause I did a lot of coding even before I took these java classes, I did a significant amount of coding in c++, some of these ones are harder to learn than others, and usually you start with Java because the APIs are so massive, and you can basically do everything with it if you make calls but if you haven’t coded at all before, I can see where it just gets so frustrating that you want to just leave, because If you leave out a double equals somewhere, you are having a full true statement every single time, and you can’t catch it I mean even I when I had those errors and I am looking through a lot of code for like a microcontroller And then I finally see it because if you’re using a C compiler it would tell you..so that can be like “Yeah why I am I doing this” “Why is this my minor?” It just becomes really frustrating and You can feel like “ I really just want to stop at this”.

Researcher: and I was thinking like, because I am researching introductory programing courses, sometimes that kinda becomes an important role right like you’re responsible
for them to not think that computer science is hard, and it just not your thing you know... anyway, last question. Have you ever had an experience when you did not get the help from a TA and you're just like Chuck this, I will just... Has that happened?

Participant 1: yeah, for my EE class, I have gone there and then they have basically told me which registers I needed to change for a certain thing and I have done it already and since they are on a time crunch, I have been sitting there for 30 minutes, waiting for a specific question and just basically receive no help at all, So.. I am not gonna go back into the queue, that's just.. I will go ask a friend and try to get a better answer from someone else or just work on the problem myself because it's not worth trying to take up their time again for them to give me another you know bad question sometimes, it's just not worth it but I mean if you have all your stuff there you can just work on the problem there and sit in queue and try to elicit an answer but it's just really discouraging to go to office hours when if you have put off an assignment, I think that is the biggest thing, is that I personally don't find it worth it even going to the TAs or their office hours in general if I have just spent too much time and put it off too long, there's just no point..

Researcher: In CS thought, has that happened that you just couldn't get?

Participant 1: Yeah, I had to ask around in the office and just ask people if they have seen a person because it was just my second time being there and then I like, they have very specific rooms, I think 2114 is like one room in the back corner and so it took me a bit to figure that out and so I am asking for a TA around for 2114 till someone actually says “all the 2114s are over there in that corner”, it feels like there should be like a little paragraph or 2 on.. like you guy's should have a TA website or something like that, or just a website for how the administration should work in that room and how the general structure is like if there is certain classes be in different places, because it's more of like a free for all, there are just tables, people sit wherever they want. I assume that GTAs for certain classes, if a table is open they will just sit in that table and if other tables are taken, there is no real structure, you just go and hopefully you will recognize the TA you have had or, the TA of your lab is there and then you ask them cause then you know them personally, otherwise it could be like a search.

Researcher: what about a time when you actually asked, got the TA asked your questions and still got like no help, has that happened?

Participant 2: it's happened to me, so there's been a time when I have gone and haven't been able to find the TA and then I leave. And then there's been another time when I have gone in and you wait in a long line and you finally get to the TA and then they make a comment about what is going wrong and then the comment is either not pertaining to your problem like they thought it was or you can decipher from their hint or what you should be changing or how you should be changing it. And at that point you are recycled back into the mix of people to wait in line and so, at that point you just leave.
Researcher: so let’s assume that there was not a big line of people but then you were there and then the hint that was given to you was not necessarily making you understand or get you any closer to the solution, what would you do then.

Participant 2: ask for a follow up, I mean that would be great, because then I can ask them to elaborate a little bit, like: maybe If I tried this would this work?”, if you don’t really have that back and forth ability with the instructor.

Participant 1: if the basic interaction doesn’t help you, the you try to get some more back and forth because some of these answers to the question can literally just be something so.. I mean if it’s something so little that you have already googled that and seen that you kind of need some direction beyond just “oh this error pertains to that”, it’s probably one of your printf calls or something, I have looked through it, I know what that means, I want someone to look at my code and really be like “Oh, I see it you need to look at this part” to definitively tell you, because you can look at a line of code that has an error in it, if you don’t know it has an error in it, you will just read through it and go to the next stuff, I mean if your compiler is not catching it or it’ getting you a general error, which is the worst, I mean you can read over an error multiple times, it’s not going to help you at all, you need someone who can pick out that error and see where it is specifically and then tell you where it is within a class.

Researcher: Okay, that’s all he questions I had.. Thank you so much for all your input and time.
Focus group - Part 2

Researcher: Alright. So I am going to be asking you guys a bunch of questions, feel free to answer them in detail, with examples if you could think of any, based on your 1114 or 2114 experience.

Participant 3: what about twenty one fourteen?

Researcher: yeah.. 1114 or 2114 or both if you have taken both those courses. So when do you go to a TA for help? like at what point in the process do you go to a TA for help?

Participant 3: so for 1114, in lab sessions and also during, like a couple of days, before the project, you know pretty much like that.

Participant 4: It depends, if the problem is with logic, I’ll usually try to figure it out myself, for at least you know, an hour, and then If I can’t I will ask for help, if it’s syntax, I’ll ask immediately, maybe I will try, I will look it up and if I can’t figure it out in one minute, I will ask for help.

Researcher: so it kinda depends on what it is that you are trying to achieve?

Participant 4: Yeah, I mean, I assume that logic I can always figure it out myself but Syntax it’s just more efficient to ask for help in that case, you know? with logic like, it’s sometimes less efficient because you have to explain everything, the other TA they have to read everything you wrote and sometimes it’s not obvious what you are trying to do and it’s just a huge process to go through all that, well with syntax it’s like right away they know the answer usually.

Researcher: Would that also mean that, with respect to going to TA for help, you kinda want it to be quick? is that sorta implied?

Participant 4: I mean.. quick.. like I don’t mind spending a long time talking about it, as long as we are making progress you know, but with a logical issue, if you ask them a question and they have to read a bunch of code first and ultimately getting them up to speed is like the hard part right, maybe they can spot the error, maybe they can’t, there is a chance that the whole thing is just going to be like.. you know what I mean, so like it’s quick, but ultimately it doesn’t matter if it takes a long time as long as I feel like we are actually making progress on the issue as opposed to just having them read the code you have written so far, which sometimes just takes forever, you know?
Researcher: so how often do you seek a TAs help? like in a semester let’s say..

Participant 3: honestly, I would prefer everyday..

Researcher: and why is that?

Participant 3: I like solving problems of my own but sometimes I just get stuck, like specific area of study. So in this case I would seek TAs help I mean they’re very helpful on these kinda stuff.

Researcher: So when you are stuck somewhere.. you?

Participant 3: Yeah, I just prefer like I can ask a question everytime if I want to..

Participant 4: Yeah, I usually ask questions like couple of times every lab, for the projects much less common, but you know, we worked on one together right the HashMap, that was stuff that was using a lot of Concepts that I was not familiar with, so I just came to the office hours to talk to you about it, but for all the other projects, I basically just wrote it on my own but in Lab, because there is a limited time, basically every time, I did not want to fool around, trying to figure out everything by looking it up myself, I just wanted to know the answers so I could finish in time..

Researcher: Okay.. Have you ever stopped yourself from going to a TA for help? and if yes, why was that?

Participant 3: I don’t know.. honestly, Personally I prefer solving problem on my own but sometimes, I can’t like quite figure out some like some specific part like in 1114 session, I had some problem with my exam, since I didn’t have a good grade on my exam, I had to seek the correct answer for the problems, so I don’t think I can solve this problem on my own so, I seek for help..

Researcher: But have you ever said I won’t go to a TA this time, I will just do it on my own or find some other way to do it? has that happened?

Participant 3: I think it happens.. I want to solve this problem on my own, I want to think like I can solve this kinda problem, without seeking TAs help..

Researcher: So in your case, I guess it would be, if there is a TA I would go ask even if..?

Participant 3: I would try to avoid asking so many questions, like unnecessary questions, but sometimes I just need help on some specific part..

Researcher: Have you chosen like a different method maybe, like not go to a TA but maybe post on the class forum..or ??

Participant 3: I google a lot..
Researcher: (to other participant) Have you stopped yourself from going to a TA?

Participant 4: Only because it is inconvenient to come here, so I don't ever mind asking a question if I am in lab and you (TA) are sitting right there, it's always fine, but if I am at home, it's like I could go to office hours tomorrow but then I have to come all the way here and just that's when I would ever not do it.

Researcher: any other reason??

Participant 4: No, I mean, TAs are a good resource, I am all about utilizing that resource, It's just that, sometimes it's just not worth it, you know what I mean, so...like for bigger things, the HashMap example again, I knew I was going to have more than one thing to ask, you know, I could tell that it was going to be, because I also couldn't remember how to use scanners and import files and all that stuff, so that specific time, I knew that there was going to be more than one question, I didn't want to start emailing do all the stuff. I would rather just sit down and like do it, you know, so I made sure that there was a time I could come and see you...but for small things like again, it's usually just better to figure out yourself than...

Researcher: Right. Okay...what is the kind of help that you expect when you go to a TA? What do you expect from them? Like I would like it if they help me this way or a different way you know?

Participant 3: Ahmm.. It depends on the question, sometimes I prefer an approach to this kind of specific question, So I can solve it during the exam, and for my projects, like such kind of a thing, I prefer a quick answer like what should I do to correct my error or something like that.

Researcher: would you happen to have an example like to differentiate the two?

Participant 3: Oh yeah..like during exam, we will have some conceptual questions and this kinda question requires a lot of understanding on the coding like the knowledge behind it, so this kinda question or knowledge, we are supposed to have a deeper understanding on it so I would prefer an approach of how do I understand this concept, for the project, is just like more specific questions, So I really need some specific answer for it..

Researcher: okay okay, would you like it if they told you how to fix it or would you...?

Participant 3: If it is specific, I can accept it..

Participant 4: I prefer just the direct answer, because honestly for me I am always going to ask more questions if I don't understand you know, so if you say something that to me doesn't seem to fit with what I have learned already with Java, then I am always going to ask a follow up question because like I understand that ultimately if you see a similar question you have to know how it works, like just knowing one answer
is not going to help you. So I really just prefer the most straightforward answer possible, and then it will be up to me if I want to know more.

Researcher: what about the approach? I mean, what if they choose to sort of just lead you and you find it out yourself?

Participant 4: I mean I know that like teaching theory says that that's the best way to do it, but I think that's the best way for the average student that needs like a little more guidance on like how to learn. But I think that I don't need that. So it's kind of. For me it's not necessary, and is a little bit irritating sometimes,

But at the same time, I mean you know I don't like, it's not like upsetting to me, it doesn't like me not want to seek the TAs help n the future, it's just one of those thing where it's like I know we have to do this dance and it's just what it is you know.

Researcher: Okay. Okay... would you prefer to go to a Particular type of TA, or a set of TAs maybe, if you had like a choice?

Participant 3: yes, I do..

Researcher: and that would depend on?

Participant 3 I used to very be enjoying harassing [TA X], ( group laughs ), one time, like during thanksgiving break, we didn't have a mandatory lab session but I still come to lab session, so I can finish the lab, something, I don't remember the number, He is a pretty good instructor like, to tell me how to approach this kinda question, he won't just give me like a specific answer for that but he leads. Sometimes it's very frustrating, because I don't know what he is talking about and most of the time it's very confusing, but after I solve the question, I feel very accomplished.

Researcher: oh.. Then the struggle becomes more like “oh that's why I did all this”, when you actually like get to the point..okay. So would you say that you started preferring to go to his office hours to get his help because ?? like was there a reason that you started doing that or was it just like ?

Participant 3: I like his teaching style. And he has like.. Hmm.. he has some sort of deep understanding on this kinda concept, specific question..

Participant 4: well the only TAs I had for CS where you and [TA Y] right, you guy's are both good, so [TA Y], I only have experience working with her in labs a few times, and I mean she pretty much knew the answer to the questions that I had in that caseand they weren't very deep questions tanyway so you know, and then with you like, the only time we really got to work together was the hashmaps stuff right, I appreciated the fact that you took time to actually write the exception to like suppress that error that we where getting for the import from the different not the student.io thing, so that was nice.. But I mean I haven't had like lotsa experience, so it's hard for me to say like who I would prefer, you know.
Researcher: Let's assume you had a group of TAs, you know like in the form that you filled out, if you had like a choice, would you prefer to go to a certain type of TA?

Participant 4: I mean when you say certain type of TA, you mean like from the choice that you listed on there, it was like the type of answer they are likely to give you...as I was saying before I mean ultimately I want the most direct and specific answer possible because I will always ask more questions if that doesn't answer the entire question I have you know, I would rather be it on me to lead myself further in the understanding than have somebody who is trying to like hold your hand through that process, you know what I mean, like I said before also I understand why that is the default because a lot of students are not like that and ultimately you are here to teach and not just feed people answers that they can move on to the next step you know.

Researcher: How is getting help or learning from a TA different than learning in lecture or getting help from peers? or other methods like discussing on an online forum. Or google for that matter?

Participant 3: Usually during lecture our instructor will teach us a lot of different stuff you know in a very condensed time, so in the meantime, we might have like some certain understanding issues so for this case we have to ask like TAs to solve a part of a puzzle because if you don't like understand a certain part of lecture, in next lecture it will be very confusing like because the.. Like the.. You don't have a deep understanding of the previous lecture..

Researcher: and then there is dependency. Right..

Participant 4: For 1114, I did not think that the lectures were particularly well organized honestly, and also the class was huge, not that many people asked questions, so it was never an issue to ask a question, but I just didn't think that it was very streamlined which I mean I don't understand why that's the case because it's literally the first class we ever take in the CS course, so it's been taught here a 100 times, and like you would think it would be a well run machine, but it really wasn't, you know there was a lot of times like I just didn't feel like the lecture was really as useful as it could have been and it could have been a lot more organized honestly, but that being said, that is also kinda how a lot of those big lectures are kinda like that sometimes... were the professor is just like they give you the basics and expect you to kinda learn on your own after that but I dunno...

Researcher: based on that, how do you like think that TAs bridge that? the big class to the small..

Participant 4: The TA is kinda like the recitation part of the lecture right, so anything that you have specific to ask, that's where you would prefer to do it because you don't necessarily take a lot of class time asking about a specific example of something, and ultimately it's not just that one question usually, you have that and then answer spawns another one and like you really want 5-10 minutes, whereas in a class that is 50 minutes long, taking up ten mins for a particular question, that's really not relevant to the rest of the class is not, you know it's not ideal.
Researcher: that's the inherent disadvantage of a big class right?

Participant 4: yeah. I mean generally I think a lot of people just don't ask a question at all because they don't want to ask a question in front of a 100 people and sound stupid, I don't have that problem, but i think people do..

Researcher: does it feel any different when you are getting help from a peer or?

Participant 4: it depends on how good the peer is right? If he is very smart, then it can just be like getting help from a professor, but as far as asking questions on the forum or going online, like going online is great, if it works right away right, but if it doesn't it's frustrating, posting on the forum is great because you almost always get the answer to your question, but asking follow ups like there is a delay there which is unsatisfying because a lot of times you are trying to finish a project and you have other stuff to do and you are waiting ten minutes to go back and forth for something that could be accomplished in like one minute talking to somebody face to face but that being said, you can ask those questions from your house, so it's always a trade off right?

Researcher: okay, is there any difference between talking to a TA during office hours and talking to them during lab, like assuming that those are the only times you get to interact with the TAright? like face to face at least.

Participant 3: it doesn't make a difference..

Researcher: really?

Participant 3: No..

Researcher: Okay..

Participant 4: I agree it doesn't make a big difference, the only question is how many people are waiting for your attention right, so in office hours, there are a lot of people there then it's the same as in lab if there is a lot of people waiting,, the only difference in lab is that everyone need to finish by a certain deadline, so like the pressure to have the TAs attention is a lot higher versus office hours, assume that you are not there at the last minute before the project is due or something so you don't feel as bad about taking up a lot of time or waiting for a long time, like obviously sometimes that's not true but n lab it's always true that everyone needs to finish..

Researcher: apart from the time art of things, is there anything else that's different in a teaching style for a TA like in lab vs TA in office hours.

Participant 4: not in my experience..

Researcher: Have you ever had an experience when You did not get the help from a TA?
Participant 3: I do. Like one time I was doing my project, I was trying to find a TA and I found one and she was quite irresponsible.

Researcher: In the sense that...?

Participant 3: in the sense that, she did not solve my problem and she just left.

Researcher: okay, because it was end of time?

Participant 3: ah.. Like pretty close.

Researcher: why is that irresponsible?

Participant 3: because I was trying to ask her a question, it wasn't pretty hard, it would have just taken a couple of minutes for it, and she like read through the whole thing and she made some very confusing comments on it, and that was it..

Researcher: So she did respond but she did not actually like stay to finish it?

Participant 3: nope.

Participant 4: I don't have such experience, no.

Researcher: the mirror of that question, have you ever felt encouraged or motivated specifically after receiving help from a TA? this might be with respect to your 1114, maybe just when you were struggling with the initial phases..(long pause) did [TA X] ever motivate you?

Participant 3: could you repeat the question again..?

Researcher: sure,..have you ever felt particularly encouraged or motivated after receiving help from a TA?

Participant 3:.. Not specifically..

Researcher: but do you think that, that could be something that the TAs could do? Like you might have not felt very, you might have not felt that 1114 is hard or 2114 is hard, but there might be other students who might feel that way right, do you think that TAs can actually help in that front or?

Participant 3:.. It really depends..like it depends on the student, from the student perspective, they might find it very helpful, they might find it like.. it depends on the students motivation..

Researcher (to the other participant): your thoughts on that?
Participant 4: for me, not really, but I am already a highly motivated student so I don’t think really, but I mean I guess if there is one thing that I could say that like anytime you are looking at a problem, that particular thing if there is more stuff that you could suggest to try or do or something like that, I am just like grabbing at straws here, because I don’t like have a good example, but I mean like I have worked as a tutor before and for very highly motivated students, people that wanna go further in the field I like to suggest things like, “well if you think this is interesting then you should look at this because what you are doing right now is kind of existing in vacuum, but “ (this is for math, particularly right), “but you will see if you go further in this field, you are going to see all sorts of application to this that some of them are really going to blow your mind because like if you think this is powerful, like wait till you see the real big machinery that uses this underlying network, right?”

Researcher: It’s like triggering the interest more..

Participant 4: Yeah, I mean the idea is just like this is what ultimately, like we are not here to learn this, we are here to like memorize this stuff and apply it to big high level problems right...this is the doorway into that.. Something that you could read or look at whatever but that being said like again, I already have that so I don’t think that I was missing it..

Researcher: that’s all the questions I had.. Thank you very much for your time.. I appreciate it.